Grau en Enginyeria Informatica

Plataforma cloud d'exercicis de ciberseguretat

Ian Sangines-Uriarte Muñoz

Juny 2018

Resum

Aquest projecte tracta de la implementació d'una plataforma per donar suport als cursos de seguretat informàtica que dóna el Laboratori d'Innovació InLab FIB, concretament a la part pràctica d'aquests cursos. A la part pràctica, els alumnes apliquen la teoria apresa durant el curs amb diverses màquines virtuals, unes intencionadament vulnerables que contenen ciberexercicis i d'altres amb diferents eines que s'acostumen a utilitzar per fer anàlisis o auditories.

L'objectiu principal és implementar una plataforma *cloud* que contingui totes les màquines virtuals necessàries per al curs i diversos procediments per incorporar-ne de noves, tot de forma centralitzada, i on cada alumne disposi dels recursos necessaris per utilitzar les màquines que li facin falta. D'aquesta manera els professors estalviaran temps en preparar l'entorn per a cada alumne i els alumnes podran fer la part pràctica des de qualsevol lloc.

Resumen

Este proyecto trata de la implementación de una plataforma para dar soporte a los cursos de seguridad informática que imparte el Laboratori d'Innovació InLab FIB, concretamente de la parte practica de estos cursos. En la parte practica, los alumnos aplican la teoría aprendida durante el curso con varias maquinas virtuales, unas intencionadamente vulnerables con ciberejercicios y otras con varias herramientas que se suelen utilizar para hacer análisis o auditorías.

El objetivo principal es implementar una plataforma *cloud* que contenga todas las maquina virtuales necesarias para el curso y varios procedimientos para añadir nuevas maquinas, todo de forma centralizada, y dónde cada alumno disponga de los recursos necesarios para utilizar las maquinas que les hagan falta. De esta manera los profesores ahorraran tiempo en preparar el entorno para cada alumno y los alumnos podrán practicar des de cualquier sitio.

Abstract

This project deals with the implementation of a platform to support the cybersecurity courses taught by the Laboratori d'Innovació InLab FIB, specifically the practical part of these courses. In the practical part, the students apply the theory learned during the course with several virtual machines, ones intentionally vulnerable with cyberexercices and others with several tools that are usually used for analysis or audits.

The main objective is to implement a cloud platform that contains all the necessary virtual machines for the course and different procedures to add new ones, fully centralized, and where each student has the necessary resources to use the machines that they need. In this way, teachers will save time in preparing the environment for each student and students will be able to practice from anywhere.

Agraïments

Donar les gràcies al director del projecte Antonio Rodríguez i a la cap de departament Antonia Gómez per facilitar-me aquest projecte i supervisar-lo durant el seu desenvolupament.
Especials agraïments a Daniel Sánchez per ajudar-me a trobar solucions a tots els dubtes i problemes durant el transcurs del projecte.
I per últim, però no menys important, agrair a la meva família i companys la seva paciència i el suport que m'han donat durant el projecte i des de sempre.

Índex

1 Introducció

Aquest projecte és un Treball Final de Grau de l'especialitat de Tecnologies de la Informació de la Facultat d'Informàtica de Barcelona (Universitat Politècnica de Catalunya). Es tracta d'un projecte de la modalitat B desenvolupat en el laboratori d'innovació InLab FIB, amb l'objectiu de donar suport a cursos de formació en ciberseguretat que ofereix el laboratori tant a profesionals com a estudiants.

1.1 Contextualització

A mesura que creixen les tecnologies i n'apareixen de noves, la seguretat pren cada cop més importància i hauria de ser un dels principals requisits en la implementació de qualsevol projecte.

A causa d'aquest creixement, cada cop es generen més dades que poden ser compromeses i caure en males mans, vulnerant així la nostra privacitat. Existeixen patrons d'atacs informàtics[1] per extreure informació de bases de dades, per accedir al sistema de fitxers, per robar credencials del navegador web, etc. Però els problemes de seguretat no només afecten les nostres dades, sinó també als sistemes que hi ha a la xarxa. Es poden produir atacs DoS (de l'anglès *denial-of-service*) i fer caure el servidor evitant així poder servir les peticions,o es pot segrestar una màquina i obtenir el control total i remot d'aquesta i d'altres en la xarxa.

La millor manera d'evitar aquests atacs és sent conscient de per on et poden atacar i securitzar al màxim els teus sistemes i dispositius, però com ja sabem, ningú neix ensenyat. Per aprendre a securitzar un sistema primer s'ha de saber com atacarlo, per així conèixer les vulnerabilitats que té i com es poden explotar.

Però com aprens a atacar un sistema si no en disposes de cap? Doncs aquest projecte és el que intenta solucionar, oferint un entorn controlat amb diferents màquines, eines i sistemes vulnerables amb els que jugar i aprendre atacant.

1.2 Formulació del problema

En la formació que imparteix l'InLab FIB de ciberseguretat, la part pràctica del curs necessita un entorn per fer els exercicis. Aquest entorn el prepara el professor amb les eines o els sistemes necessaris per fer les pràctiques sobre el temari del curs, que normalment inclou diversos sistemes vulnerables, un sistema operatiu amb les eines

per poder fer els atacs i material complementari. Tot aquest entorn està pensat per poder-se desplegar mitjançant màquines virtuals en una màquina física i local per cada alumne del curs, però aquest procés té bastants inconvenients.

Primerament, les màquines físiques necessiten certs recursos per poder executar l'entorn, ja que per cada exercici són necessàries diverses màquines virtuals actives amb el consum de memòria i de processament que això comporta. Per fer-nos una idea, la màquina local hauria de disposar d'uns 8GB de memòria RAM i un processador mitjanament potent, com per exemple un Intel Core i5 i així aconseguir un entorn que funcioni de manera fluida. Com que la formació s'acostuma a donar en les instal·lacions del client, no sempre es podrà disposar d'una màquina amb aquests requeriments per cada alumne.

Un altre problema amb què ens trobem és el desplegament de l'entorn. En el cas que el curs s'imparteixi amb màquines físiques del client, el professor ha d'instal·lar el programari necessari per executar l'entorn en cada una de les màquines de les quals es disposen i desplegar-lo. Aquest procediment es fa força pesat tenint en compte el nombre de màquines físiques necessàries i les dificultats que es puguin trobar en el procés d'instal·lació de l'entorn.

Finalment, en el pitjor dels casos, serien els alumnes qui portarien les seves pròpies màquines físiques per fer el curs. Això implicaria que cada alumne s'hauria d'instal·lar l'entorn en la seva pròpia màquina i seria el professor qui s'encarregaria de fer arribar el material necessari per instal·lar l'entorn a cada alumne. D'aquesta manera cada alumne, amb els recursos que disposa la seva màquina, ha d'esperar que li arribi el material i fer la instal·lació en horari del curs, pel que es perd força temps i més si es donen problemes en la instal·lació.

Per donar solució a tots aquests problemes, s'ha implementat una plataforma amb OpenNebula[7], un orquestrador de màquines virtuals que treballa com a plataforma de computació en el núvol, de manera que ens permet desenvolupar un entorn distribuït amb tots els exercicis de ciberseguretat necessaris per fer la formació, que es desplegarà en un servidor prou potent en el que els alumnes hi podran accedir remotament i fer la part pràctica sense cap requeriment addicional (a part d'un navegador) en les màquines físiques que utilitzin.

1.3 Objectius del projecte

L'objectiu d'aquest treball és oferir una plataforma *cloud* que contingui diverses màquines virtuals amb exercicis de ciberseguretat per donar suport a cursos de formació de ciberseguretat que realitza l'InLab FIB per a qualsevol nivell i disciplina. Aquesta plataforma ha de poder permetre a l'alumnat instanciar o apagar màquines a la seva voluntat per poder així posar en pràctica la teoria que es dóna al curs. El professor de la formació ha de poder donar d'alta els usuaris amb els quals els alumnes accediran a la plataforma, administrar els permisos d'aquests usuaris per gestionar les màquines necessàries per dur a terme la formació i poder introduir noves màquines per a futurs cursos.

Per tal de poder assolir aquest objectiu, es duran a terme diferents tasques:

Configurar i integrar una aplicació web

Configurar un servidor per allotjar les màquines virtuals

Configurar, definir i introduir les màquines virtuals a la plataforma

Junt amb el professor de la formació, s'haurà de triar quins exercicis es faran servir i penjar-los a la plataforma. Cap la possibilitat de crear-ne de nous per adaptar-los a alguna formació en concret.

1.4 Actors implicats

Les parts implicades o interessades en aquest projecte es poden diferenciar en cinc grups:

InLab

Els implicats/des d'aquest grup són Antonia Gómez (responsable de l'àrea de sistemes de l'InLab) que ha definit i dissenyat el projecte segons els requisits necessaris per a la implementació, Antonio Rodríguez (Director del Projecte) que ha recollit els requisits necessaris per al projecte i que alhora serà el professor del curs,el ponent Manel Medina i finalment la part administrativa de l'Inlab que, gràcies a aquest projecte, podrà oferir cursos amb una nova plataforma.

Client

El client que contracti el servei de formació en ciberseguretat serà també beneficiari d'aquesta plataforma amb la que els alumnes del curs podran practicar i formar-se.

Professor

Aquesta plataforma ha estat pensada per facilitar una part de la feina del professor, per tal que imparteixi la formació de ciberseguretat de manera més senzilla i eficient.

Alumnes

Els alumnes de les formacions utilitzaran aquesta plataforma per aplicar els coneixements teòrics adquirits en la formació sense haver de preocupar-se de implementar un entorn de pràctica.

Desenvolupador

El desenvolupador s'encarrega de portar el projecte al dia, dissenyar-lo i especificar-lo. També s'encarrega de documentar i desenvolupar el projecte, així com de preparar el treball a entrgar, escriure'l i presentar-lo.

2 Estat de l'art

2.1 Ciberseguretat

Amb l'aparició de la Internet i els sistemes connectats a la xarxa, tots aquests sistemes podien ser vulnerats i s'havien de securitzar. S'havia d'anar molt amb compte, si es volien trobar vulnerabilitats, d'atacar el sistema que estava en producció, ja que es podia caure el servidor i deixar de servir les peticions dels clients. L'única manera de securitzar-lo era amb un entorn controlat de proves. Replicant el servidor i desplegant la mateixa plataforma en local, aquesta es podia vulnerar sense cap problema, perquè si queia, es podia reiniciar el servidor i seguir fent proves. Aquest mètode tenia molts inconvenients, com per exemple aconseguir un servidor el més semblant possible al de producció, trobar lloc per ubicar-lo, configurar la xarxa per poder-hi accedir, proveir-lo d'energia, etc. Tot això comportava un cost en temps i diners.

Més tard, gràcies a l'aparició de les màquines virtuals, es va poder aplicar una solució més eficient. Per vulnerar o auditar el teu sistema només feia falta instanciar una màquina virtual en qualsevol màquina local i així obtenir un entorn controlat per atacar-la. Fins i tot, gràcies a la facilitat de gestionar aquestes màquines virtuals, és possible representar una xarxa real per així auditar-la i conèixer les seves vulnerabilitats. L'únic desavantatge que té aquesta solució respecte de l'anterior és els recursos necessaris que ha de tenir la màquina local on es desplegarà la màquina virtual. Hem de pensar que és un sol ordinador que està executant dos sistemes operatius (com a mínim) i podria ser que no fos prou potent.

No només per auditar el teu sistema sinó per aprendre sobre ciberseguretat, aquesta solució ens ho posa molt fàcil, per exemple, creant màquines virtuals per practicar. Seria el cas de màquines virtuals com Badstore[2] o Web for Pentester[4], que són màquines virtuals que contenen una aplicació web expressament vulnerable per a qui vulgui aprendre a hackejar pàgines web. Existeixen també màquines virtuals com Metasploitable [3], que contenen un sistema operatiu amb serveis vulnerables, per aprendre a utilitzar eines que permeten explotar aquestes vulnerabilitats.

Però, i si es dóna el cas que la nostra màquina no és prou potent per poder aixecar aquestes màquines? La solució a aquest problema és la implementació d'un sistema distribuït.

Amb el concepte del *cloud*, s'han implementat moltíssimes plataformes amb funcionalitats que només podíem tenir en els nostres ordinadors però en sistemes distribuïts, com per exemple sistemes de fitxers (Dropbox), editors de text online (Google Drive), etc., que es poden accedir en remot des del navegador.

En el cas de la ciberseguretat també ha estat així. Existeixen plataformes d'aprenentatge com HackThisSite[5], que hosteja diferents pàgines web vulnerables a les quals es pot accedir per practicar. Una altra plataforma més complexa és HackTheBox[6], que conté diverses màquines virtuals vulnerables que es poden instanciar a gust de l'usuari per fer els exercicis que et proposis.

Aquest projecte vindria a ser un d'aquests tipus de sistema distribuït però adaptat únicament per la formació de ciberseguretat que s'imparteix a l'InLab FIB, amb els requeriments demanats per l'InLab.

2.2 Eines avaluades

Per assolir els objectius d'aquest projecte, es necessita una eina que ens faciliti la feina a l'hora de gestionar i executar les màquines virtuals. Un dels primers passos del projecte és decidir-se per una d'aquestes eines, i per fer-ho s'han hagut d'estudiar les diferents alternatives i veure si encaixaven amb l'especificació del projecte. Aquest estudi s'ha dut a terme llistant les avantatges i les desavantatges de cada alternativa. Seguidament s'exposen les eines estudiades:

OpenStack[8]

És un software *Open Source* de virtualització per crear *clouds* tant públics com privats.

Funciona de manera que diferents projectes (o serveis), proporcionats pel mateix software i connectats entre si, construeixen una *IaaS* (Insfrastructure as a Service) que permet crear una plataforma per implementar i gestionar nodes de computació, nodes d'emmagatzematge i recursos de xarxa virtualitzats a través d'un panell de control web accessible, alhora, pels usuaris i per l'administrador.

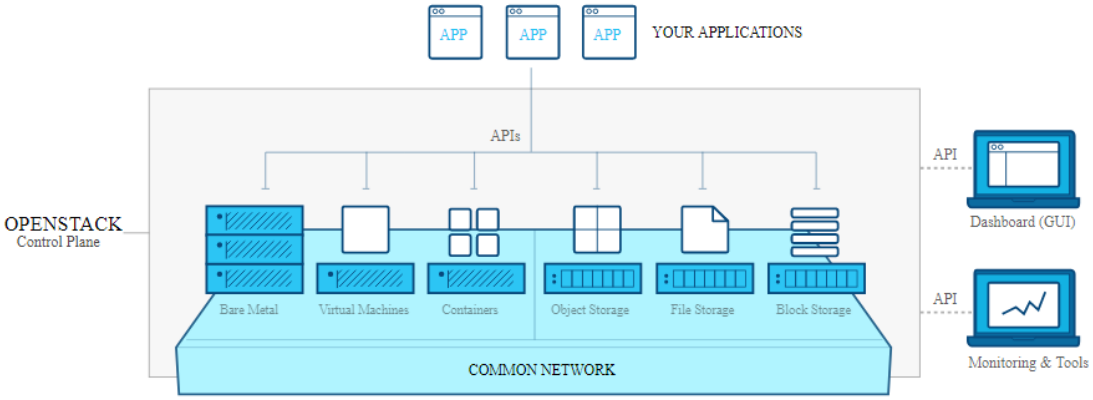


Figura 1: Arquitectura d'una instància Openstack i els seus serveis

Avantatges:

- Al estar format per diversos projectes o serveis fa que sigui una plataforma molt modular i, per tant, molt escalable.
- En ser un software *Open Source* hi ha una comunitat molt gran darrere que participa en el seu desenvolupament i que pot oferir ajuda amb qualsevol problema a l'hora d'utilitzar-lo.

Desavantatges:

- Està format per diversos projectes o serveis que el fa menys manejable i dificulta la configuració
- Els requisits hardware per a implementar-ho són força elevats.

vSphere[9]

És un software de virtualització de tipus 1 (on el software s'executa directament sobre el hardware i no sobre un sistema operatiu) propietat de VMWare.

Està format per un sistema operatiu anomenat ESXi, que s'encarrega d'executar les màquines virtuals i que conté eines de gestió i administració d'aquestes. La plataforma vSphere inclou també un software per fer la gestió més amigable amb una interfície gràfica accessible per l'administrador.

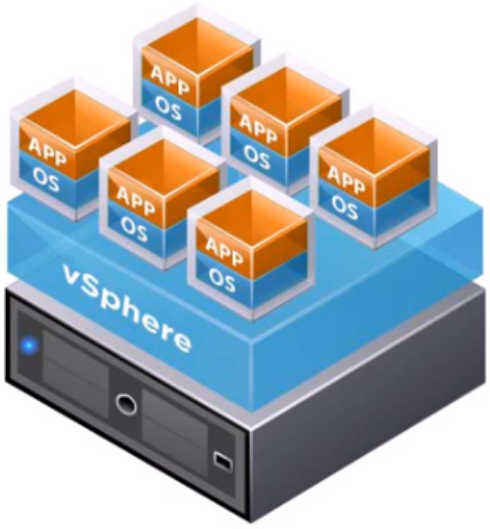


Figura 2: Arquitectura hypervisor vSphere

Avantatges:

- És un sistema operatiu molt lleuger (144MB) basat en GNU/Linux que es pot instal·lar en qualsevol màquina.
- La seva arquitectura és molt senzilla i no necessita dependències d'altres softwares, de manera que es pot configurar molt fàcilment.

Desavantatges:

- És un software de pagament, tot i que té una versió gratuïta però amb moltes menys funcionalitats.
- No té cap panell de control pels usuaris, només per a l'administrador.

OpenNebula[7]

És un software *Open Source* que funciona com a orquestrador per gestionar i adrministrar *clouds* tant públics com privats.

Utilitza eines de virtualització ja existents i proveeix una capa superior que permet crear i gestionar nodes de computació, d'emmagatzematge i recursos de xarxa virtualitzats a través d'un panell de control web accessible, alhora, pels usuaris i per

l'administrador.

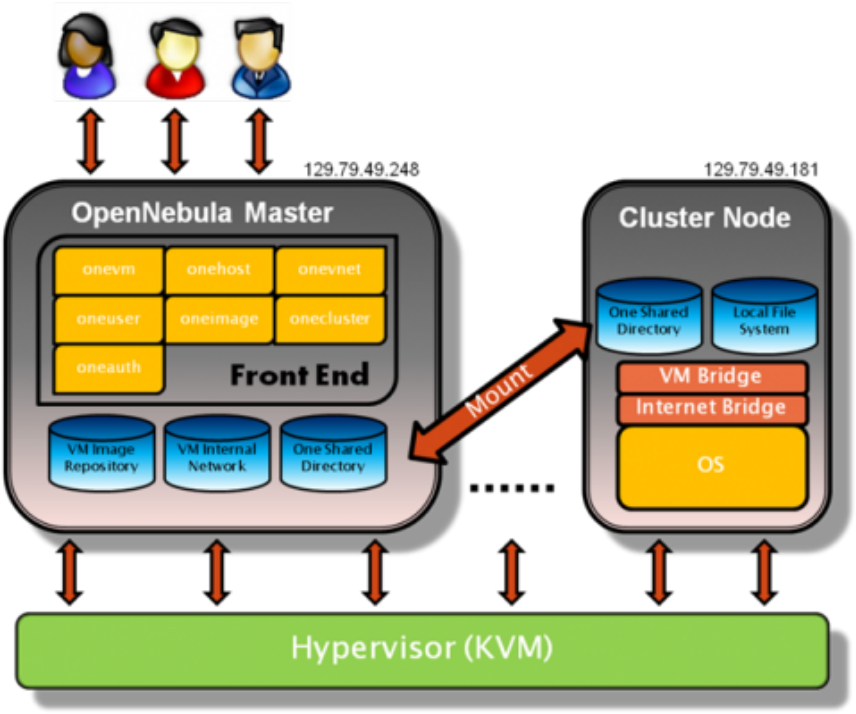


Figura 3: Arquitectura d'un sistema OpenNebula

Avantatges:

- En ser un software especialitzat per a ser utilitzat com a orquestrador fa molt fàcil la configuració de la mateixa plataforma.
- En ser un software *Open Source* hi ha una comunitat molt gran darrere que participa en el seu desenvolupament i que pot oferir ajuda amb qualsevol problema a l'hora d'utilitzar-lo.

Desavantatges:

- Necessita un *hypervisor* amb el que integrar-se per poder executar les màquines virtuals.
- Els requisits hardware per a implementar-ho són força elevats.

Docker[10]

És un software *Open Source* que permet executar aplicacions dins d'un sistema de manera aïllada mitjançant contenidors, proporcionant així una *Paas* (Plataform as a Service).

En aquests contenidors es poden desplegar les mateixes aplicacions que en un sistema operatiu, pero la gestió dels recursos la fa el mateix sistema Docker amb les funcionalitats que proveeix el sistema operatiu.

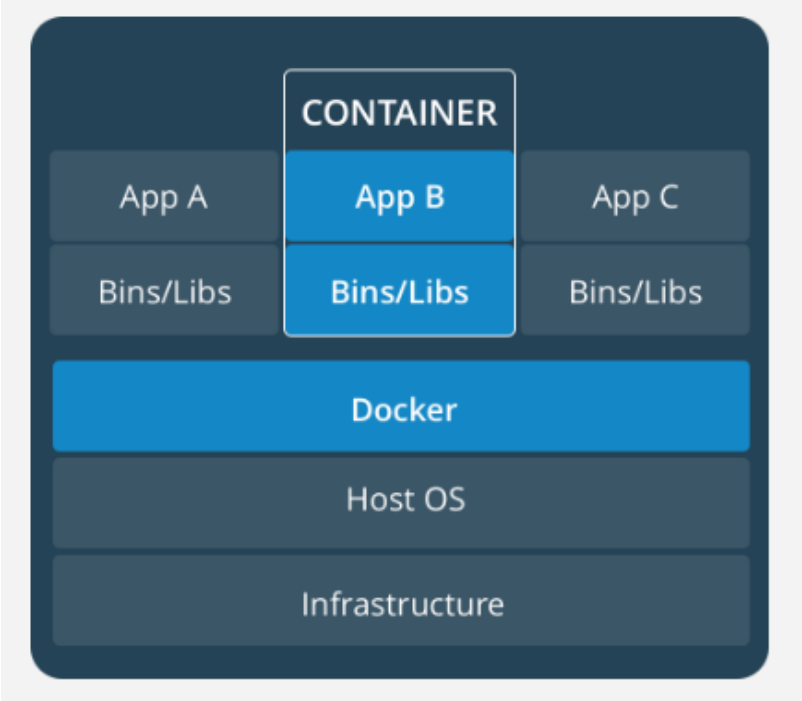


Figura 4: Arquitectura contenidors de Docker

Avantatges:

– Permet crear contenidors molt lleugers i executables per a qualsevol màquina.

– A diferència de les màquines virtuals, els contenidors no necessiten cap *hypervisor* per a executar-se, sinó que s'executen sobre el mateix sistema operatiu

Desavantatges:

– No té cap eina de gestió amb interfície gràfica per als contenidors.

– No permet executar sistemes operatius, només aplicacions.

2.3 Conclusió

Després d'haver especificat el projecte i discutir les diferents alternatives, hi havia dues eines sobre la taula que ens podrien servir, Openstack i OpenNebula.

Es va descartar ESXi perquè no té cap mena de front-end per la gestió de les màquines per part de l'usuari, només poden ser gestionades per l'administrador.

Es va descartar Docker perquè fa molt difícil l'execució d'un sistema operatiu amb GUI, ja que només permet instal·lar aplicacions amb un sistema operatiu especialment preparat per executar-se amb Docker. Tampoc conté cap mena de front-end per a l'usuari, tot i que en aquest cas seria més fàcil implementar-lo que amb ESXI.

De les dues alternatives restants ens hem decidit per **OpenNebula** , ja que per l'especificació del projecte i els seus requeriments era l'opció més adient. OpenNebula fa més fàcil la gestió de les màquines i la configuració de la plataforma, a diferència d'Openstack, que és una plataforma més complexa i més potent del que es necessita.

Finalment, una altra part important de la decisió venia donada per l'experiència dins de l'InLab amb OpenNebula, gràcies al fet que ja s'havia implementat alguna plataforma amb aquesta eina.

3 Abast

3.1 Delimitació de l'abast del projecte

Per assolir l'objectiu del projecte, les parts a desenvolupar són les següents:

Configuració i integració amb una aplicació web

Aportarà una manera amigable per poder gestionar les màquines virtuals i instanciar-les quan sigui necessari. Tots els participants de la formació faran servir aquesta web:

- El professor serà l'encarregat de donar accés als alumnes a aquesta plataforma perquè puguin instanciar les màquines necessàries per fer la formació amb els permisos corresponents.

- Els alumnes podran instanciar i apagar les màquines que utilitzin per la formació i accedir a la interfície gràfica de la màquina virtual per Computer Security Incident Response Team (CSIRT).

Aquesta tasca també inclou posar en marxa un servidor per desplegar l'aplicació web i integrar-la amb el servei o API del servidor que contingui el hypervisor

Configuració d'un servidor per executar les màquines virtuals

Aquesta tasca es durà a terme en diferents fases:

- Fer recerca per trobar un hypervisor que s'encarregui d'executar les màquines virtuals sobre el sistema operatiu del servidor i instal·lar-lo.

- Fer recerca per trobar un orquestrador que es pugui integrar amb l'hypervisor i que ens permeti gestionar les màquines virtuals fàcilment.

- Implementar una API o servei que comuniqui l'aplicació web amb l'orquestrador que s'hagi triat prèviament.

Preparació dels exercicis i integració amb la plataforma

Amb l'ajuda del professor, fer una tria de les màquines necessàries per a les formacions i penjar-les a la plataforma.

3.2 Possibles obstacles

Aquests són els possibles obstacles que ens podem trobar durant el desenvolupament del projecte:

Desconeixement de la tecnologia

Al no tenir coneixements gaire avançats en virtualització i no haver treballat mai amb un hypervisor requerirà molt de temps adquirir coneixement i resoldre els problemes que es puguin donar.

Infraestructura

Seran necessaris dos servidors per dur a terme el projecte, un dels quals necessita molta potència de computació a l'executar les màquines virtuals. Si aquest servidor no és prou potent pot dificultar i alentir les proves i el sistema en si.

Complexitat de les màquines

Adaptar les màquines perquè es puguin executar en l'entorn que es vol desenvolupar pot ser una tasca molt costosa. Un altre aspecte a contemplar és el desconeixement de les màquines, que podria complicar encara més el procés i donar problemes de compatibilitat amb el sistema.

4 Metodologia de treball

4.1 Mètodes de treball

Per a la realització d'aquest projecte s'ha utilitzat una metodologia de tipus àgil.

La metodologia àgil consta de processos anomenats sprints on s'especifica, junt amb el client, i es desenvolupa una part del projecte en un determinat temps (aproximadament 2-4 setmanes). Passat aquest sprint en ve un altre en el qual es desenvolupa una altra part o funcionalitat del projecte. Aquesta metodologia s'acostuma a aplicar en equips de desenvolupadors per acotar la seva feina a un període de temps i perquè el client obtingui resultats en cada sprint.

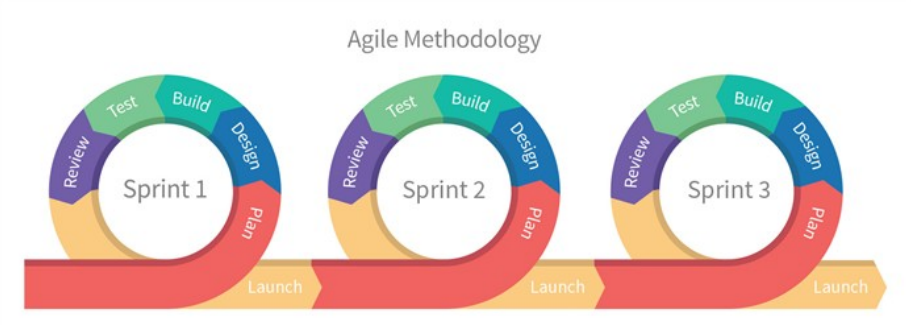


Figura 5: Esquema de desenvolupament amb metodologia àgil

En el cas d'aquest projecte, la durada de l'sprint no depenia del temps, sinó dels requisits o funcionalitats a desenvolupar, de manera que cadascun tenia una durabilitat diferent. Per a cada sprint, s'especificava amb el client (en aquest cas el professor de la formació) què era el que necessitava per aquella part del projecte. Seguidament es feia recerca sobre com solucionar el problema i s'implementava. Un cop implementat, es feien proves sobre el sistema i es passava al següent sprint si no hi havia cap error.

4.2 Eines de seguiment i validació

La gestió, el seguiment i la validació del projecte es feia setmanalment amb el director del projecte de manera presencial. En el cas del seguiment presencial amb la cap del departament, es feia mensualment.

De manera no presencial, el seguiment del projecte es feia mitjançant l'eina Trello, on hi ha constància de l'especificació de tot el projecte i de les tasques que esta fent el desenvolupador en cada moment.

Pel seguiment del projecte com a desenvolupador s'ha fet servir l'eina Google Drive, on s'han documentat diàriament els avenços en el projecte i tots els problemes que s'hagin pogut donar durant el desenvolupament.

Per a la comptabilització d'hores s'ha fet servir l'eina Toggl, una plataforma web que permet fer el recompte de les hores per dia i dividir aquestes hores en diferents projectes o tasques.

5 Planificació

Al ser de modalitat B, aquest projecte requereix unes 735 hores aproximadament. Vist aquest requisit i treballant a mitja jornada, el projecte es desenvoluparà durant 8 mesos en 20 hores setmanals. El projecte començarà l'Octubre del 2017 i acabarà el Juny del 2018.
Donat que el treball està realitzat només per una persona, totes les tasques que es duguin a terme seran seqüencials, tot i que hi ha alguna tasca que depèn del departament de sistemes de l'InLab.

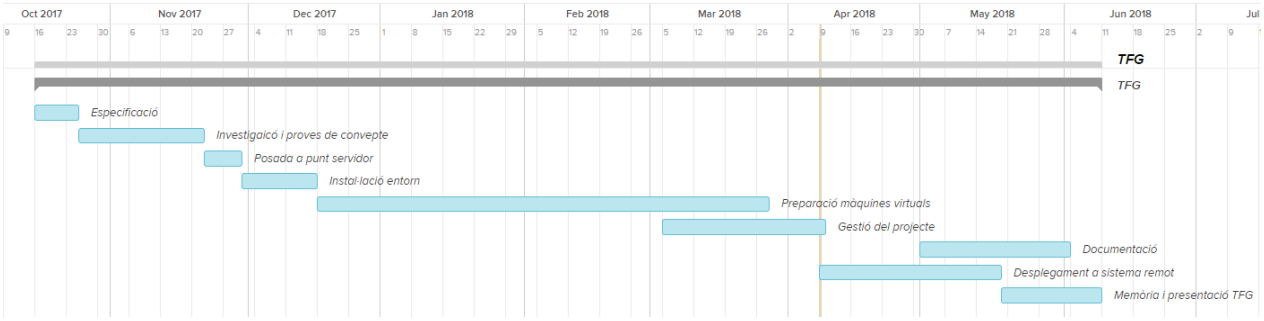
5.1 Temps requerit

Aquest treball s'ha desenvolupat amb un total de 784 hores distribuïdes de la manera com indica la taula 1.

Tasca	Hores
Especificació del projecte	28
Investigació i proves de concepte	80
Posada a punt dels servidors	24
Instal·lació entorn	44
Preparació Màquines virtuals	244
Desplegament màquines a Microsoft Azure	112
Documentació	92
Gestió del projecte	100
Memòria i presentació TFG	60
Total	**784**

Taula 1: Taula amb el temps corresponent per tasca

5.2 Diagrama de Gantt



5.3 Descripció de les tasques

A continuació es descriuran les principals tasques del projecte i les seves dependències.

5.3.1 Especificació del projecte

Aquesta és la primera tasca a fer en el projecte. Tracta d'analitzar els requisits i objectius del projecte..

Per dur a terme aquest procés es faran reunions amb la cap de departament i el professor de la formació (que també és el director del projecte) en el període d'una setmana. Mentre, individualment, es valorarà el contingut de les reunions i es començarà a investigar com desenvolupar el projecte.

5.3.2 Investigació i proves de concepte

Aquesta tasca depèn directament de la tasca anterior 5.3.1

Feta l'especificació del projecte amb els requisits i els objectius definits, és hora d'informar-se sobre tecnologies i documentar-se.

En aquesta tasca s'investigarà per trobar la millor solució per desenvolupar un sistema cloud de màquines virtuals. S'haurà de trobar una plataforma que ens permeti virtualitzar qualsevol màquina i que ens ofereixi alguna eina per accedir via web a les funcionalitats de gestió de les màquines virtuals. Es llegirà documentació sobre els orquestradors més utilitzats i es faran proves de concepte per entendre el seu funcionament i veure si s'adapta a les necessitats del projecte.

5.3.3 Posada a punt dels servidor

Aquesta tasca no depèn de cap altra tasca del projecte, però sí del departament de sistemes, que és qui proporcionava els servidors necessaris per a la plataforma.

Un cop proporcionat l'espai als servidors físics, s'haurà d'instal·lar el sistema operatiu a cadascun. Feta la instal·lació, s'hauran de configurar per incloure'ls a la xarxa de l'InLab i per securitzar-los.

5.3.4 Instal·lació entorn de l'orquestrador OpenNebula

Aquesta tasca depèn directament de la tasca anterior 5.3.3

En aquesta tasca s'haurà d'instal·lar l'orquestrador OpenNebula en els servidors i configurar-lo perquè funcioni correctament. Aquest procés consistirà a allotjar l'aplicació web que ens proporciona la plataforma en un dels servidors i la instal·lació i configuració de l'orquestrador en l'altre.

5.3.5 Preparació màquines virtuals

Aquesta tasca depèn directament de la tasca anterior 5.3.4

Després d'haver instal·lat l'entorn, s'hi pujaran les màquines virtuals amb els ciberexercicis. Aquest procés és el més complicat, ja que s'han de tenir en compte moltes variables com l'espai a disc, la capacitat de processament del servidor i la compatibilitat de versions entre els sistemes operatius de les màquines virtuals i la plataforma on s'executaran.

Primer de tot s'haurà d'aconseguir suficient espai per poder emmagatzemar les màquines virtuals, i després s'hauran d'adaptar les màquines perquè puguin executar-se en aquest servidor. Aquesta última tasca traurà molt de temps en el projecte, ja que l'adaptació de les màquines es basa en prova i error, de manera que cada pas que es faci per a la preparació s'haurà de provar a la plataforma. En el cas que no funcioni, s'haurà de revertir el procés i pensar alternatives per fer l'adaptació.

5.3.6 Desplegament màquines a Microsoft Azure

Aquesta tasca depèn directament de la tasca anterior 5.3.5

A l'haver comprobat que totes les maquines preparades funcionen i es poden executar en la plataforma, hem de configurar-la per a que s'executin en un sevrei extern (Microsoft Azure). Aquests serveis ens ofereixen capacitat de computació i servidors virtualitzats per a poder executar les màquines virtuals amb els recursos necessaris.

5.3.7 Documentació

Aquesta tasca depèn directament de la tasca anterior 5.3.6

Un cop el sistema estigui a punt per funcionar i abans d'abandonar el projecte com a desenvolupador principal, s'ha d'escriure la documentació del projecte.

Aquesta plataforma haurà de ser ampliada i mantinguda, s'hauran de preparar més màquines, pujar-les a petició del professor i possiblement canviar la ubicació de la plataforma a servidors més potents. De manera que el següent desenvolupador que es trobi amb el projecte necessitarà saber tots els procediments que s'han fet per construir la plataforma, per preparar-la i per fer-la funcionar.

5.3.8 Gestió del projecte

Aquesta tasca no en depèn de cap altre.

En aquesta tasca es redactarà i es prepararà tot el contingut de l'assignatura de GEP.

5.3.9 Memòria TFG

Aquesta tasca depèn de la tasca anterior 5.3.8 i de la tasca 5.3.7

En acabar el projecte i l'assignatura de GEP, es realitzarà l'última tasca que consistirà a redactar la memòria i preparar la presentació del TFG.

5.4 Recursos utilitzats

En aquesta secció s'enumeraran els principals recursos utilitzats en aquest projecte.

5.4.1 Recursos humans

Per aquest projecte l'únic recurs humà és el desenvolupador, que hi estarà treballant a temps complet les hores pertinents al dia desenvolupant i gestionant el projecte.

Encara que hi hagin participat més persones com el Director de projecte o les del departament de sistemes, es comptarà com a recurs principal únicament el desenvolupador.

5.4.2 Recursos hardware

- Servidor de la FIB DELL PowerEdge R510 amb sistema operatiu ESXi de VMWare i una màquina virtual Ubuntu 14.04 LTS amb 1GB de RAM i 50GB de disc per allotjar la web.
- Servidor DELL PowerEdge R510 amb SO ESXi de VMWare i dues màquines virtuals Ubuntu 14.04 amb 32 GB de memòria i 8 CPUs per executar les màquines virtuals de la paltaforma.
- Ordinador de sobretaula HP propietat de l'InLab FIB amb Intel i5, 8GB de RAM i Windows 10 per a desenvolupar el projecte

5.4.3 Recursos software

- Editor de text OverLeaf per escriure la memòria en Latex
- Google Drive per portar dia a dia la gestió del projecte i les tasques fetes.
- Trello per a escriure l'especificació del projecte i fer el seguiment amb la Cap de departament i el Director del projecte.
- Redmine per documentar el projecte per l'InLab
- hypervisor KVM per poder virtualitzar en Linux

5.4.4 Recursos cloud

- Microsoft Azure per executar les màquines virtuals en un sistema distribuït i aconseguir el recursos necessaris per a les màquines.

5.5 Valoració d'alternatives i pla d'acció

Com en qualsevol projecte, es podrien donar incidències o desviacions que no permetessin avançar en el desenvolupament o que fessin perdre part del projecte ja desenvolupat.
Gràcies al fet que el projecte ha durat 8 mesos, s'ha pogut permetre assumir alguna incidència i poder acabar el projecte en el temps estimat.

Seguidament es mostra una taula amb els possibles incidents i les solucions adequades:

Incidència	Gravetat	Solució
Desconeixement de la tecnologia	Mitja	Documentar-se i fer proves de concepte per familiaritzar-se
Caiguda de xarxa o del sistema	Greu	Com que no es poden fer Backups dels servidors utilitzats per manca d'espai, s'haurà de documentar extensament el procés d'instal·lació del sistema
Problemes amb els servidors o mala gestió per part del desenvolupador	Greu	Com que no es poden fer Backups del projecte per manca d'espai, s'haurà de documentar extensament el procés d'instal·lació del sistema
Incompatibilitat amb les màquines virtuals vulnerables i OpenNebula	Mitja	Pel fet que poden haver-hi problemes amb les versions de les màquines virtuals, la solució més fàcil és substituir la màquina per una altra.

Taula 2: Taula amb les possibles incidències i com solucionar-les

5.5.1 Desviacions produïdes

Finalment, s'ha pogut acabar el projecte sense incidències i amb el temps establert. Tot i així, el temps reservat a la resolució de contratemps s'ha aprofitat per investigar funcionalitats extres de la plataforma OpenNebula que no eren necessàries en aquest projecte.

6 El projecte

6.1 OpenNebula

Tal com han definit els objectius del projecte, es vol implementar una plataforma on poder executar i accedir a diverses màquines virtuals que contenen exercicis de ciberseguretat. Després d'haver valorat diferents opcions per fer-ho (com mostra l'apartat 2.2) ens hem acabat decidint per la solució que ens dóna OpenNebula.

OpenNebula és un software *open source* que fa la funció d'orquestrador i que per tant, ens permet administrar centres de dades virtualitzats públics, privats o híbrids.

En ser la infraestructura totalment independent de la plataforma, fa el projecte molt escalable, deixant-nos augmentar el nombre de màquines i d'usuaris en un futur millorant la infraestructura.

6.1.1 Components OpenNebula

OpenNebula consta de dos components principals per poder utilitzar la plataforma:

OpenNebula Sunstone

Aquesta part d'OpenNebula està formada per una pàgina web i una base de dades. La pàgina web serveix per administrar les màquines virtuals, els usuaris, la xarxa i tots els paràmetres de la plataforma, mentre que la base de dades s'utilitza per persistir totes les dades necessàries i fer consistent el sistema amb els diferents components.

La primera pàgina que ens trobem és el *login*. Sempre que l'usuari autenticat estigui més de 15 minuts sense interaccionar amb la plataforma es tancarà la sessió i tornarà a aparèixer la pàgina de *login*. El temps d'inactivitat amb el que es tornarà a demanar l'autenticació pot ser configurat per l'administrador.

Gràcies al gestor d'usuaris i de permisos d'OpenNebula, poden haver-hi molts tipus de vistes i moltes combinacions d'accés als recursos d'OpenNebula, depenent de com l'administrador ho hagi configurat. En el cas d'aquest projecte, només existiran dos tipus d'usuaris:

Usuari Administrador

Aquest usuari té accés i la capacitat d'administrar totes les vistes d'OpenNebula i a tots els recursos.

Un cop dins l'aplicació trobarem el *Dashboard*, una pàgina amb la informació general d'ús d'OpenNebula.

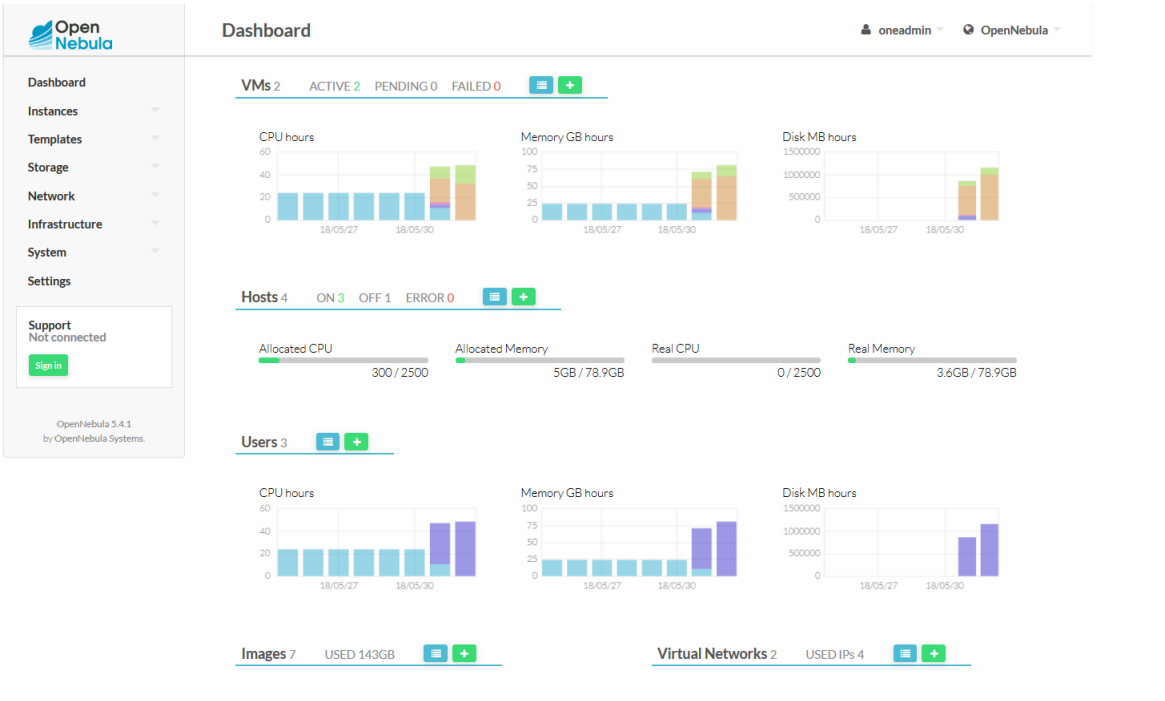


Figura 6: *Dashboard* de l'usuari administrador

Tal com es veu a la part esquerra de la imatge de la figura 6, l'administrador té accés a totes les funcionalitats que ofereix OpenNebula. Seguidament s'explicaran les més utilitzades per la gestió de les màquines virtuals i de l'entorn de virtualització.

Imatges

Dins del submenú *Storage* apareix l'opció d'*Images*, que mostra una pàgina per administrar les imatges pujades a OpenNebula. Aquestes imatges són discs virtuals preparats, tal com explica l'apartat de contextualització 6.1.2 més endavant, perquè puguin ser executats en l'entorn virtualitzat.

En aquest apartat també es poden crear imatges, clonar-les i modificar els paràmetres d'una imatge ja creada.

Plantilles

Aquesta opció mostra una pàgina per administrar les plantilles creades per l'administrador d'OpenNebula. Amb les plantilles es defineixen els paràmetres amb els quals s'executarà una màquina virtual, com pot ser el disc virtual a executar, la configuració de la xarxa, l'hotel on s'executarà i els recursos hardware.

Des d'aquesta pàgina també es permet crear, clonar i modificar plantilles i instanciar màquines virtuals a partir d'una plantilla.

Màquines virtuals

Sota el menú *Instances* trobem l'opció *VM*, que mostra una pàgina per administrar les màquines virtuals instanciades. Per cada màquina virtual descriu el seu estat i l'usuari que l'ha instanciat.

Des d'aquesta pàgina podem instanciar noves màquines virtuals, seleccionant posteriorment la plantilla que es desitgi, i ens permet també establir una connexió VNC amb una màquina ja instanciada en estat *RUNNING*.

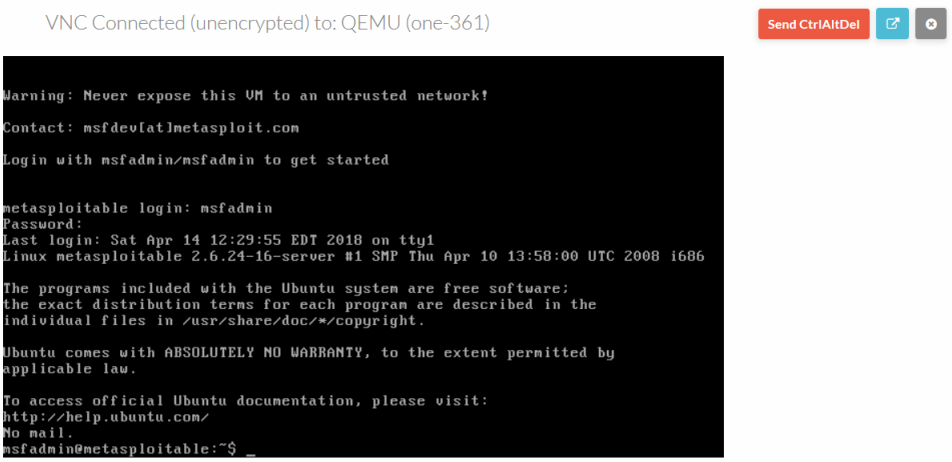


Figura 7: Connexió VNC amb una instància de la màquina *Metasploitable*

Hotels

Dins del submenú *Infrastructure* trobem l'opció *Hosts*, que mostra una pàgina per administrar els *OpenNebula Nodes*. Tal com explica l'apartat OpenNebula Node, els nodes són les màquines físiques on s'executen les màquines virtuals.

Per cada hotel es descriu el seu estat i la quantitat de recursos hardware consumits per les màquines virtuals instanciades en ells. Des d'aquesta pàgina podem crear i deshabilitar hotels.

Usuaris

Sota el menú *System* trobem l'opció *User*, que mostra una pàgina per administrar els usuaris d'OpenNebula. Per cada usuari mostra el nombre de màquines que té instanciades i els recursos hardware que està consumint. Des d'aquesta pàgina també es permet crear i modificar usuaris i només pot ser l'administrador el que els doni d'alta.

Sota l'opció *Users* trobem també l'opció *Groups*, que permet administrar els grups d'usuaris d'OpenNebula. Es pot assignar a cada usuari dins d'un grup (com si es tractés d'un sistema Linux) per més endavant poder assignar-li els permisos dins l'aplicació.

Permisos

Dins del submenú *System* trobem l'opció *ACLs*, que mostra una pàgina per administrar els permisos dins l'aplicació. Des d'aquesta pàgina també es poden crear permisos, i es poden definir per acció (utilitzar, administrar i crear), per recurs (hotels, plantilles, màquines virtuals, etc.) i per usuari o grup d'usuaris.

Xarxes virtuals

Sota el menú *Network* trobem l'opció *Virtual Network*, que mostra una pàgina per administrar les xarxes virtuals on es connectaran les màquines virtuals d'OpenNebula. Per cada xarxa virtual es mostra el nombre de màquines connectades i l'usuari que l'ha creat.
Des d'aquesta pàgina també es permet crear i modificar xarxes virtuals.

El funcionament d'aquestes xarxes i la seva configuració s'expliquen més endavant, en el punt 6.2.1.

Usuari client

Aquests usuaris només tenen la capacitat d'administrar les seves màquines virtuals. L'administrador és el que s'encarrega de preparar les plantilles

perquè l'usuari final pugui instanciar les màquines tal com necessita.

En entrar a l'aplicació també trobarem un *Dashboard*, però aquest cop reduït, només amb la informació d'ús de l'usuari.

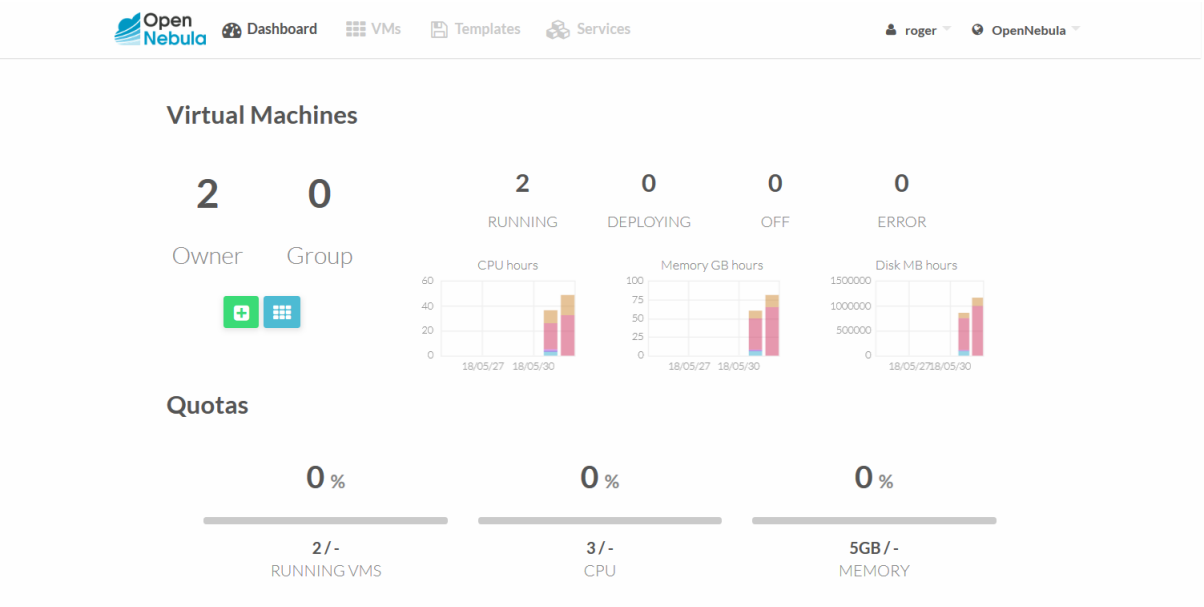


Figura 8: *Dashboard* d'un usuari client

Tal com es veu a la part de dalt de la figura 8, l'usuari només té accés a les funcionalitats d'OpenNebula que se li han permès i que s'expliquen a continuació.

Màquines virtuals

Aquesta opció mostra una pàgina per administrar les màquines virtuals instanciades. Per cada màquina descriu el seu estat i els recursos que consumeix.

La pàgina també permet instanciar noves màquines virtuals a partir de les plantilles definides per l'administrador i establir una connexió VNC amb una màquina ja instanciada.

Plantilles

Aquesta opció mostra una pàgina amb el llistat de plantilles creades per l'administrador. L'usuari només les farà servir per instanciar màquines virtuals.

OpenNebula Node El segon i últim component d'OpenNebula és la part on s'executen les màquines virtuals, el *Node*.

Per a que el node pugui virtualitzar es necessita un *hypervisor* que gestioni els recursos hardware de la màquina física, i OpenNebula ofereix integració ambdiversos *hypervisors* com KVM, VMWare vCenter, XEN i Hyper-V. En el cas d'aquest projecte, s'ha utilitzat KVM per ser una eina *open source* integrada en el kernel de Linux, que és el sistema operatiu utilitzat per les màquines que fan de node en aquest projecte, i per ser la opció amb millor integració amb OpenNebula.

A part de virtualitzar, el node d'OpenNebula rep les comandes que els usuaris fan per el *front-end* d'OpenNebula mitjançant *ssh* i s'encarrega d'executar-les i orquestrar les màquines virtuals que s'estan executant.

6.1.2 Contextualització

La contextualització d'OpenNebula és el procés en què a una màquina virtual, un cop instanciada, se li aplica la configuració adient perquè pugui ser executada en l'entorn virtualitzat i es configuren els paràmetres definits per l'administrador en la plantilla.

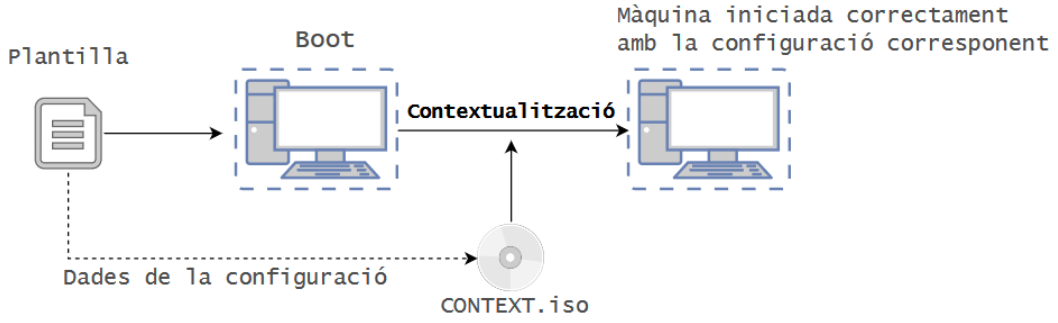


Figura 9: Esquema del procés de contextualització

Perquè OpenNebula pugui contextualitzar correctament una màquina s'ha de preparar prèviament la imatge instal·lant un paquet (proveït per OpenNebula) que

conté diversos *scripts* que s'encarreguen de la configuració de la màquina a instanciar en temps de *boot*. Aquests *scripts* llegeixen un CD virtual inserit per OpenNebula a la instància de la màquina amb la configuració definida en la plantilla i configuren la màquina virtual (tal com indicia la Figura 9).

Alguns d'aquests *scripts* s'han hagut de modificar a causa de la falta d'integració amb alguns sistemes operatius de les màquines virtuals.

```
CONTEXT = [
  NETWORK = "YES",
  SSH_PUBLIC_KEY = "$USER[SSH_PUBLIC_KEY]" ]
CPU = "2"
DISK = [
  IMAGE = "Kali - forensics",
  IMAGE_UNAME = "oneadmin" ]
GRAPHICS = [
  LISTEN = "0.0.0.0",
  TYPE = "VNC" ]
MEMORY = "4096"
MEMORY_UNIT_COST = "MB"
NIC = [
  NETWORK = "br0 - bridged network",
  NETWORK_UNAME = "oneadmin" ]
OS = [
  BOOT = "disk0" ]
```

Figura 10: Plantilla en cru de la màquina *Kali*

Per posar un exemple, partint de la plantilla de la Figura 10, els *scripts* de contextualització s'encarregarien de configurar la xarxa *br0 - bridged network*, instal·lar i configurar el servei VNC i inserir la clau pública al servei *ssh* de la instància de la màquina virtual. La resta d'apartats (OS, CPU i MEMORY) els utilitzarà l'*hypervisor* (en el nostre cas KVM) per reservar els recursos necessaris i arrancar la màquina virtual amb el disc corresponent.

6.2 Arquitectura del sistema

El sistema actualment consta de 4 servidors: Perla, Lirio, Violeta i Cactus.

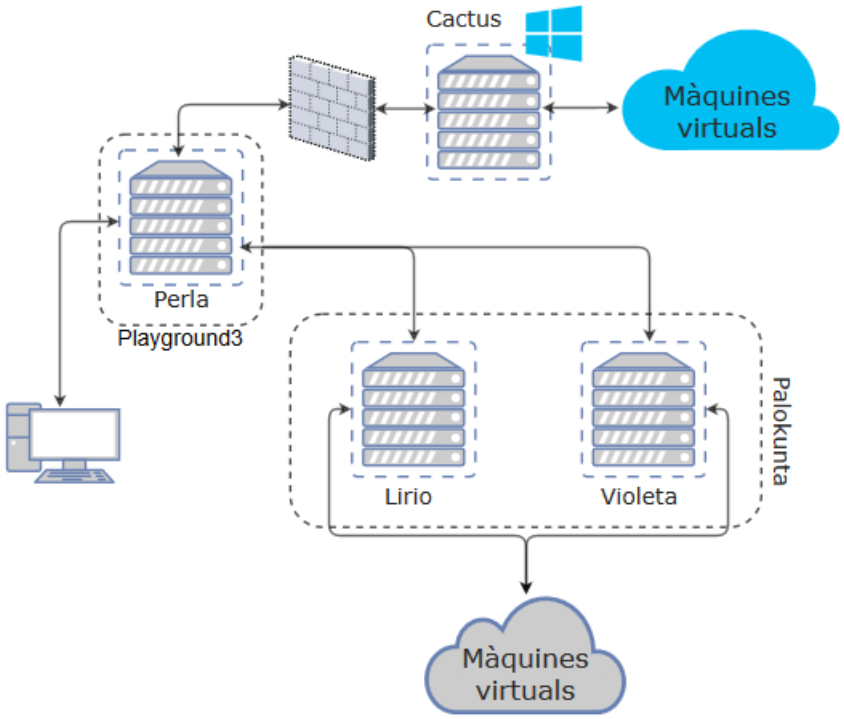


Figura 11: Esquema de l'arquitectura del projecte

Perla és el servidor que conté el front-end OpenNebula Sunstone i que es comunica amb tots els nodes per administrar les màquines virtuals.

Lirio i Violeta són els servidors encarregats d'executar les màquines virtuals. Fan la funció d'OpenNebula Node i reben les comandes des de Perla.

Cactus és un servidor allotjat a Micrsoft Azure que fa la funció d'OpenNebla Node i rep les comandes des de Perla. Aquest servidor s'ha desplegat com a prova de concepte, de manera que no té gairebé recursos i ha servit per fer un estudi sobre la viabilitat d'implementar la infraestructura OpenNebula d'aquest projecte a Microsoft Azure, tal com s'explica més endavant al punt 6.4.1.

Perla, Lirio i Violeta són servidors interns de l'InLab i s'utilitzen com a servidors

de desenvolupament i de proves en aquest projecte. Els tres són servidors virtualitzats orquestrats per VMWare i estan allotjats als servidors físics Playground3 i Palokunta.

Playground3 és un servidor compartit amb altres màquines virtuals d'altres projectes on s'allotja Perla. Al ser compartit, els recursos de la màquina són limitats, però per sort, la funció que fa Perla no requereix potència i funciona perfectament.

Palokunta és un servidor potent amb només dues màquines virtuals, Lirio i Violeta. A l'haver de virtualitzar, Lirio i Violeta han de tenir prou recursos per funcionar de manera fluida i aquest servidor compleix els requisits per fer-ho. Amb 64GB de memòria RAM i 16 *cores* les màquines virtuals per fer les proves poden executar-se perfectament.

Com ja s'ha mencionat anteriorment, OpenNebula és totalment independent a la infraestructura física que s'utilitzi. En el moment en què es vulgui fer créixer el projecte, només s'ha de configurar un altre OpenNebula Node, o els que facin falta, i comunicar-ho al front-end d'OpenNebula. Un cop els dos components estiguin connectats, OpenNebula s'encarregarà de balancejar la càrrega i executar les màquines virtuals a cada Node.

6.2.1 Configuració de la xarxa

En aquest projecte, trobem diversos components connectats entre sí amb la necesitat de comunicar-se per la xarxa.

Tal com es veu en la Figura 12, els components estan dividits en tres xarxes:

Xarxa InLab FIB

Aquesta és la xarxa de gestió dels servidors Perla, Lirio i Violeta. Mitjançant aquesta xarxa els servidors es poden comunicar entre ells i tenir accés a internet. El front-end d'OpenNebula es pot accedir des de qualsevol dispositiu a la xarxa de la FIB, mentre que els Nodes només es poden accedir des de Perla o des de l'ordinador utilitzat per desenvolupar el projecte.

VLAN88

A causa de la necessitat que hi hagi comunicació entre màquines virtuals, s'ha creat una subxarxa local virtual (VLAN), tant a Azure com a l'InLab FIB, on es connecten les màquines virtuals. Aquesta xarxa no té accés a l'exterior perquè no hi hagi possibilitat d'atacar les màquines si no és des de la plataforma

de pràctiques. Com és evident, les subxarxes de l'InLab i d'Azure no tenen connexió directa, ja que es troben en dominis diferents i l'accés a les màquines virtuals és únicament local.

Xarxa Azure

A la xarxa de Microsoft Azure és on es troba el servidor Cactus. Azure, en donar un servei IaaS (*Infrastructure as a Service*) s'encarrega d'administrar la xarxa de gestió del servidor i ens proveeix d'una màquina amb connexió a internet. Tot i això, el seu accés està restringit per un tallafoc que, com a administradors de la màquina, podem adaptar a les nostres necessitats i fer que Perla s'hi pugui comunicar per gestionar les màquines virtuals que executa.

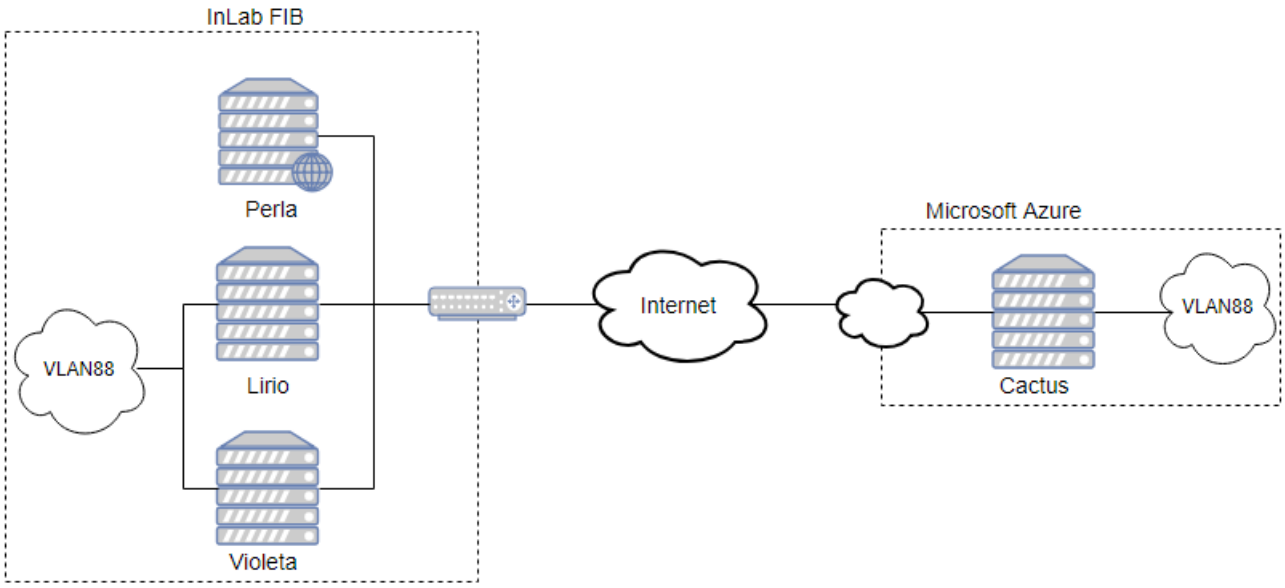


Figura 12: Esquema de la xarxa del sistema

Durant el procés de contextualització d'una màquina virtual (explicat al punt 6.1.2), una de les tasques que fa OpenNebula és configurar la xarxa tal com ho especifica la plantilla de la màquina. Per fer-ho, l'administrador crea un recurs de xarxa en el menú de Xarxes virtuals indicant el nom de la interfície del servidor on s'han de connectar les màquines virtuals i un rang d'adreces que OpenNebula assignarà a cadascuna. Però, una interfície física no és capaç de gestionar les connexions de

diverses màquines virtuals per si sola.
L'*hypervisor* KVM ens proposa una solució, creant una interfície virtual que utilitza un mecanisme de NAT i s'encarrega de traduir les adreces de les màquines virtuals a una única adreça que el router és capaç d'encaminar. Però, aquest mecanisme, per defecte, té accés directe amb l'exterior i degut al risc que comportaria per la plataforma tenir màquines vulnerables obertes a internet s'ha descartat aquesta configuració.

L'alternativa utilitzada en aquest projecte és el *bridging*. Aquest mecanisme permet assignar un *switch* virtual (br0) a una interfície de la màquina (ens34), que commutarà les connexions entre les màquines virtuals. Perquè aquestes connexions no surtin a l'exterior, s'ha creat una xarxa local virtual VLAN88 sense accés a internet on OpenNebula assignarà les màquines virtuals donant-les-hi una adreça del rang especificat per l'administrador en el recurs de xarxa.

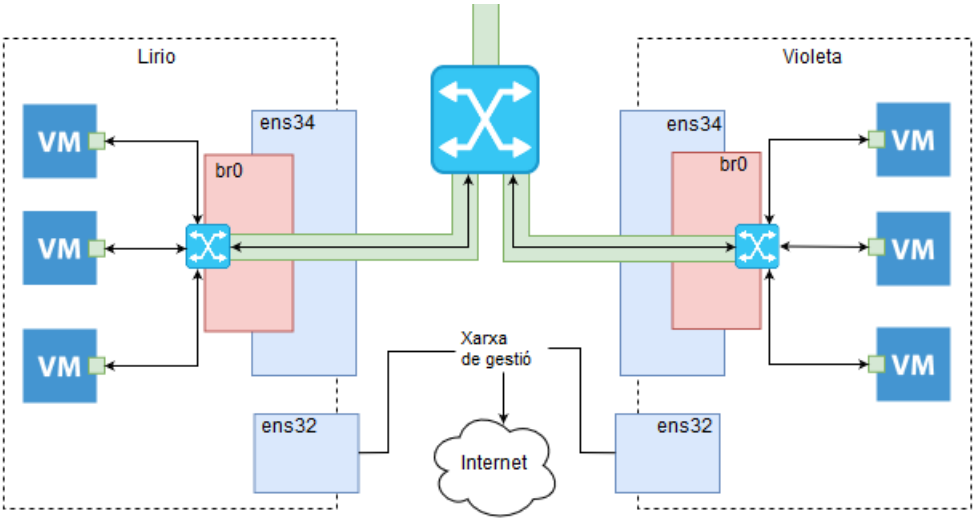


Figura 13: Esquema de les interfícies *bridge*

6.3 Ciberexercicis i imatges pujades a OpenNebula

Aquest projecte té com a objectiu donar suport a la part pràctica dels cursos de seguretat, i la manera més efectiva de practicar és atacant màquines vulnerables, però en un entorn segur.

Les màquines virtuals que s'utilitzaven en els cursos han estat preparades (contextualitzades) per poder ser executades en l'entorn d'OpenNebula. Aquestes màquines són les següents:

6.3.1 Kali Linux

Kali Linux és una distribució Linux basada en Debian orientada a proves avançades de penetració (*Pentesting*) i auditoria de seguretat. El sistema operatiu conté diversos centenars d'eines que s'utilitzen en diverses tasques de seguretat de la informació, com ara proves de penetració, investigació en ciberseguretat, informàtica forense i enginyeria inversa.

Kali Linux està desenvolupat, finançat i mantingut per Offensive Security, una empresa líder en formació en seguretat de la informació.

6.3.2 Metasploitable2

La màquina virtual Metasplotable2 és una versió intencionadament vulnerable d'una màquina amb sistema operatiu Ubuntu Linux dissenyada per provar diverses eines de seguretat i demostrar vulnerabilitats comunes.

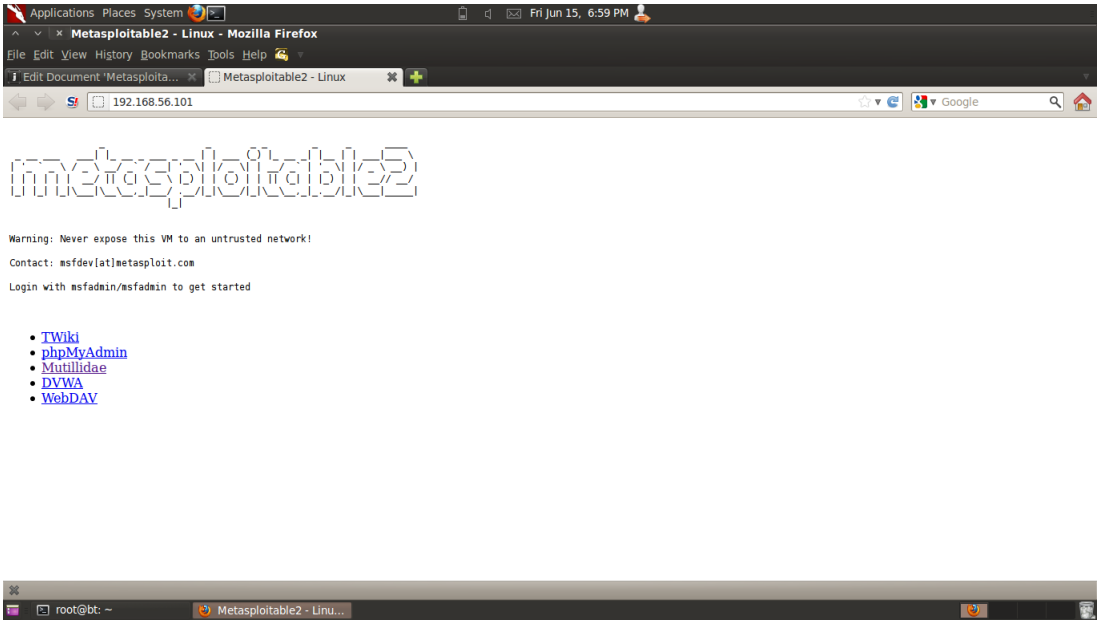


Figura 14: Pàgina web Metasploitable2

Aquesta màquina conté diversos serveis vulnerables amb els quals aconseguir accés remot a la màquina. També consta de dues pàgines web amb exercicis de ciberseguretat:

Mutillidae

L'aplicació web Mutillidae conté totes les vulnerabilitats de l'OWASP Top Ten i d'altres vulnerabilitats bastant comunes. Mutillidae permet a l'usuari canviar el nivell de seguretat de l'aplicació de 0 (completament insegur) fins a 5 (segur). A més, es proporcionen tres nivells per triar la quantitat d'indicacions donades en cada exercici, des del nivell 0 (sense indicacions) fins al nivell 2 (suggeriments màxims).

En el cas que l'aplicació quedi inconsistent per culpa dels intents d'explotació de l'usuari, el botó *Reset DB* restablirà l'aplicació a l'estat original.

Figura 15: Pàgina web Multilliade

DVWA

Damn Vulnerable Web App (DVWA) és una aplicació web basada en PHP i MySQL que té com a objectiu proporcionar un entorn per practicar algunes de les vulnerabilitats web més comunes, amb diferents nivells de dificultats, on els professionals puguin posar a prova les seves habilitats i on els desenvolupadors puguin obtenir millors coneixements en seguretat web.

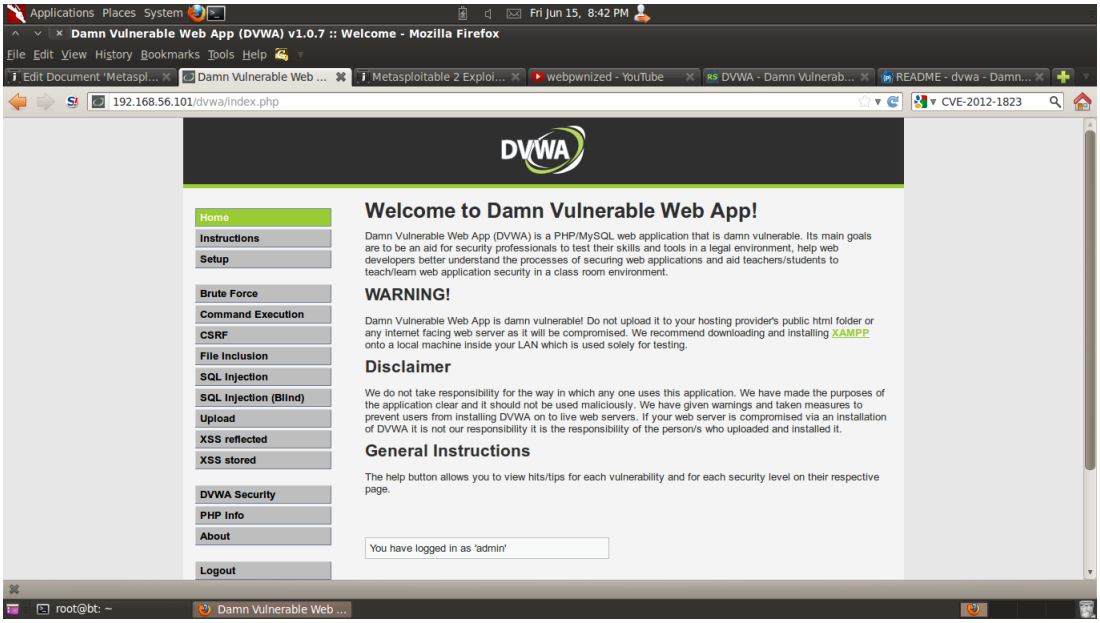


Figura 16: Pàgina web Damn Vulnerable Web App

6.3.3 PentesterLab - Web for Pentester

Web for pentester és una màquina virtual que conté una pàgina web amb diverses vulnerabilitats que podem trobar en una pàgina web.

Un cop instanciada et permet accedir a la pàgina web principal, on trobem un llistat d'exercicis on per cada *Example* augmenta la dificultat d'explotar la vulnerabilitat.

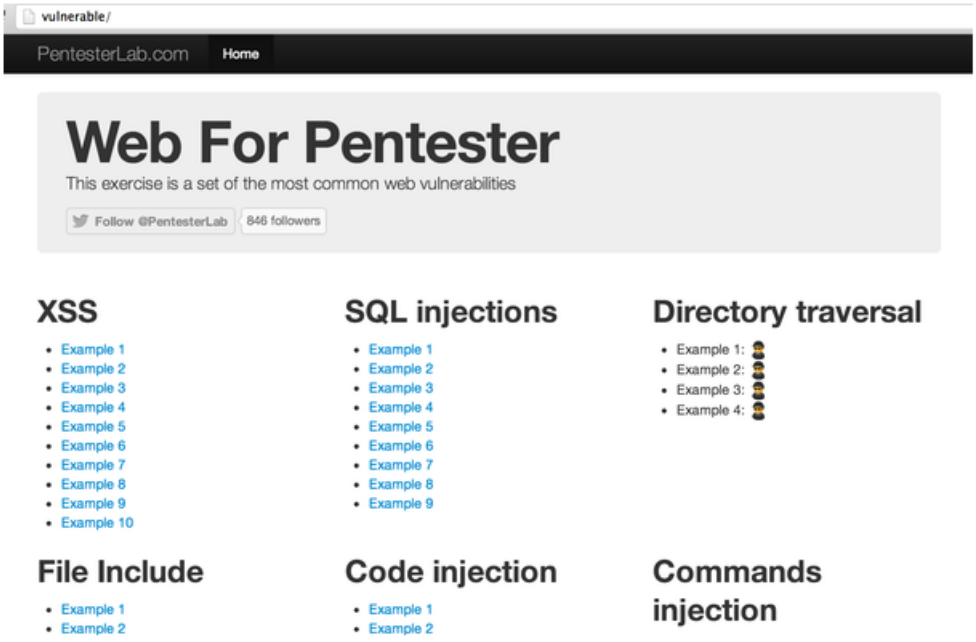


Figura 17: Pàgina web Web for Pentester

6.3.4 WebGoat8

WebGoat és una aplicació web intencionadament insegura, mantinguda per OWASP i que conté diversos exercicis de ciberseguretat.

En cada exercici, els usuaris han de demostrar la seva comprensió d'un problema de seguretat explotant una vulnerabilitat real en l'aplicació web. Per exemple, en un dels exercicis, l'usuari ha d'utilitzar un atac de *SQLInjection* (injecció de codi SQL) per robar números de targetes de crèdit falses d'una base de dades. Així, l'aplicació pretén proporcionar un entorn d'ensenyament realista, oferint als usuaris una explicació teòrica sobre com explotar cada vulnerabilitat i perquè es dóna.

Figura 18: Pàgina principal WebGoat

6.3.5 Forensics Windows 7

Aquesta màquina virtual conté un Windows 7 amb diverses eines instal·lades de *forensics* i d'anàlisis de metadades.

L'objectiu és familiaritzar l'alumne amb aquestes eines i que facin un seguit de pràctiques amb uns jocs de proves que trobaran a la mateixa màquina amb imatges de sistemes vulnerats, fitxers de logs, evidències d'atacs informàtics i recuperació de dades.

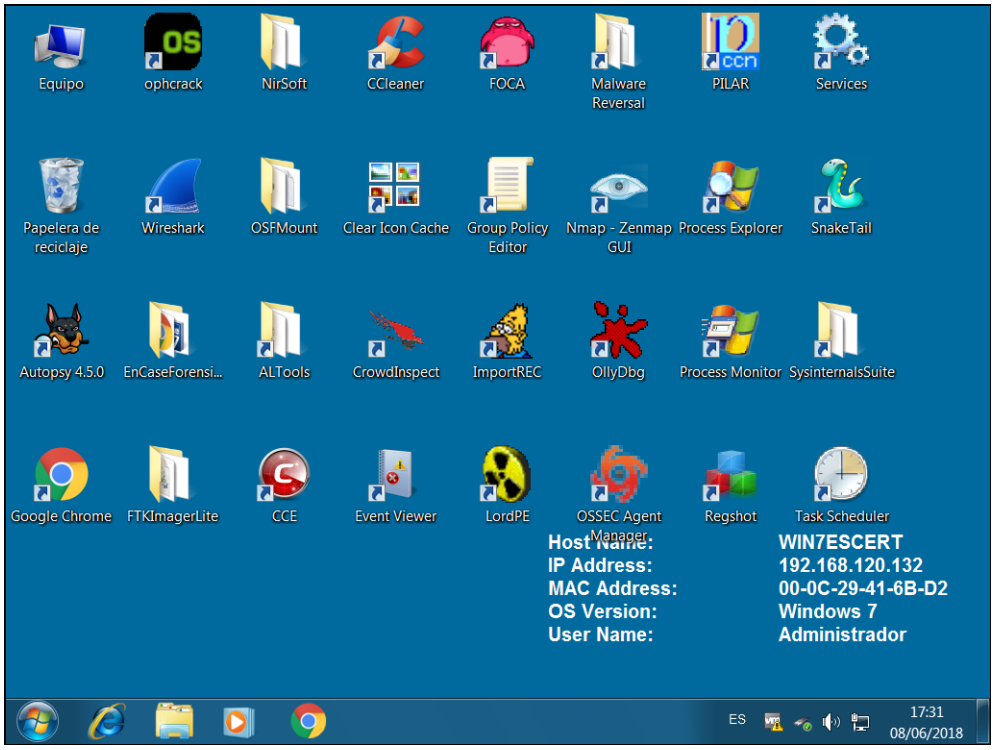


Figura 19: Windows 7 amb eines de seguretat forense

6.3.6 MISP Server

MISP és una aplicació web *open source* per recollir, emmagatzemar, distribuir i compartir indicadors de ciberseguretat, anàlisis d'amenaces i anàlisis de *malware*.

Està dissenyat per i per a analistes d'incidents, professionals TIC i de seguretat informàtica o inversors de *malware*(*malware reversing*) per donar suport a les seves operacions diàries i compartir la informació estructurada d'una manera eficient.

Les seves funcionalitats permeten l'intercanvi d'informació entre tècnics i el consum d'aquesta informació per part dels Sistemes de detecció d'intrusió a la xarxa, de les eines d'anàlisi de registres i dels SIEMs.

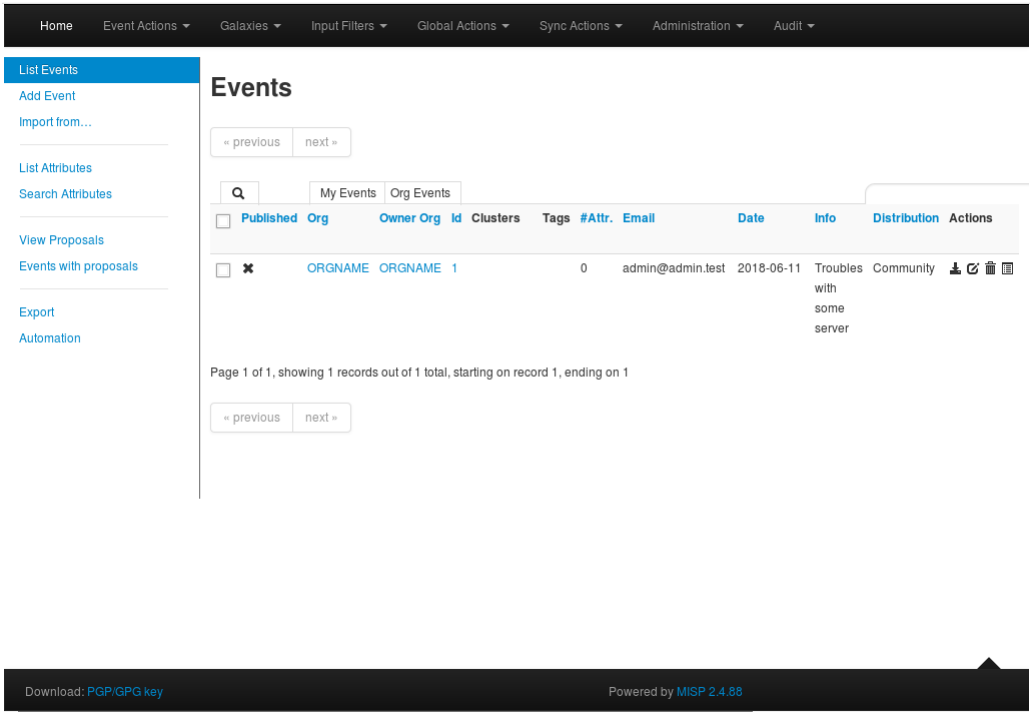


Figura 20: Pàgina principal MISP

6.3.7 RequestTracker Server

Request Tracker és un sistema de seguiment de problemes que permet fer un seguiment del que cal fer, de qui està treballant en quines tasques i de què és el que ja s'ha fet i quan. El mòdul RTIR (*Request Tracker Incident Response*) proporciona també cues i fluxos de treball dissenyats per a equips de resposta a incidents com CERTs i CSIRTs.

RTIR disposa d'eines per correlacionar les dades clau dels informes d'incidents per trobar patrons i enllaçar els informes amb la causa principal de l'incident.

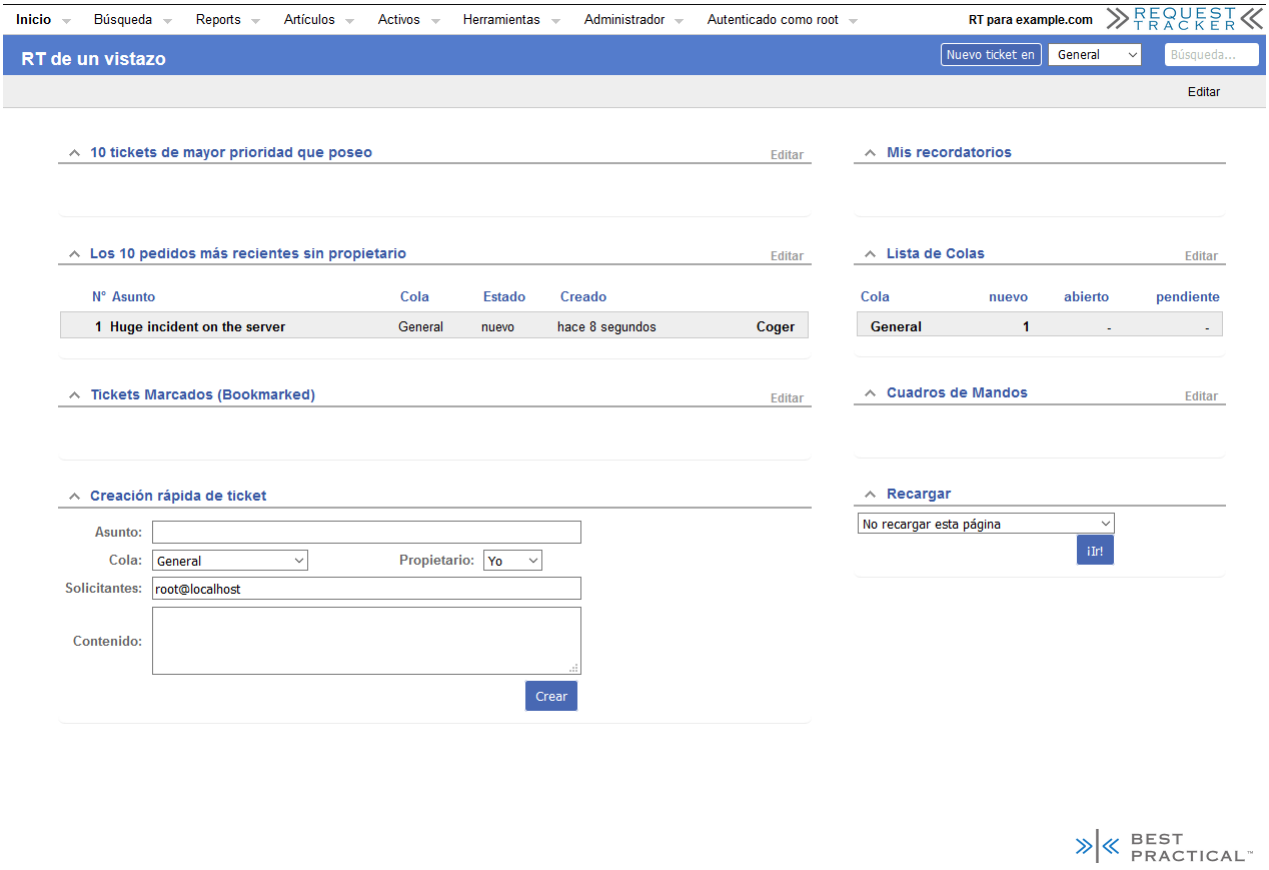


Figura 21: Pàgina principal RT

6.4 Prova de càrrega i estudi dels recursos

Un cop implementat el sistema s'ha fet un estudi per validar la seva capacitat amb la infraestructura que s'ha mencionat a l'apartat arquitectura.

L'estudi s'ha basat en una prova de rendiment simulant una sessió d'un dels cursos que dóna l'InLab FIB. En aquests cursos hi participen una mitjana de 20 alumnes que, durant la part pràctica, necessiten tenir instanciades com a mínim dues màquines virtuals, una Kali Linux i una màquina vulnerable per practicar.

Les màquines virtuals vulnerables no necessiten gairebé recursos per funcionar correctament, però el sistema operatiu Kali Linux té uns requisits mínims de 1 CPU i 1 GB de memòria (segons la documentació oficial) i uns requisits recomanats de 2 GB de memòria per funcionar de manera fluida.

Amb aquestes dades podem estimar els recursos mínims necessaris per a una sessió d'un curs:

20 Kali Linux (1 GB de RAM, 1 CPU) ⇒ 20 GB de RAM i 20 CPUs

\+ 20 Màquines vulnerables (0.5 GB de RAM, 0.5 CPUs) ⇒ 10 GB de RAM i 10 CPUs

---

Total 30 CPUs i 30 GB de RAM.

Els servidors Node de la infraestructura actual utilitzada en aquest projecte tenen un total de 64 GB de RAM i 16 CPUs. Com és evident, es fa impossible fer una sessió amb aquests recursos per la falta de CPU, així que s'ha de trobar una alternativa com Microsoft Azure.

6.4.1 Microsoft Azure

Microsoft Azure és una plataforma que proporciona diversos serveis *cloud* com ara emmagatzematge, contenidors d'aplicacions i servidors virtuals.

Existeixen altres plataformes com AWS (Amazon Web Services) o Google Cloud que ens ofereixen un servei semblant, però s'ha decidit utilitzar Microsoft Azure:

- Per la facilitat que dóna a l'hora de subscriure's

- Per la bona integració amb OpenNebula en ser una plataforma que permet el *cloud* híbrid

- Per una interfície web molt intuïtiva i amb molt bon monitoratge dels recursos

Gràcies als seus serveis, podem desplegar una infraestructura virtual que s'adapti a les nostres necessitats sense dependre de les infraestructures de la FIB. Aquesta infraestructura hauria de tenir prou recursos per executar totes les màquines virtuals d'una sessió del curs i que accepti **GlsNested Virtualization**, que és el que ens permet executar màquines virtuals en una màquina virtual.
D'entre totes les possibilitats de màquines virtuals a instanciar a Azure, s'ha escollit la *D64 v3* amb 64 CPUs i 256 GB de RAM, que té recursos més que suficients per a una sessió del curs. També ens permet virtualitzar i es podrien arribar a executar les 20 Kalis amb 2 CPUs i 4 GB de RAM perquè funcionin de manera molt fluida.

Evidentment, utilitzar aquest servei comporta un sobrecost en els cursos i estudiarem els possibles pressupostos:

Aquests cursos es desenvolupen en un total de 32 hores, que s'acostumen a dividir en 4 hores durant 8 dies.
El cost de les màquines a Azure es pot comptabilitzar per hores o per reserves d'1 o 3 anys, però descartarem del pressupost la reserva de 3 anys, ja que no es pot estimar si es seguiran fent els cursos.

Reserva d'un any
La reserva d'un any ens estalvia un 37% en el cost per hora de la màquina (1,892€/hora) i el preu mensual de la màquina és d'uns 1.381,33€, un total de més de 16.500€ l'any.

Màquina activa durant tot el curs
El cost de la màquina per hora és de 2,887€. Per aquest pressupost comptarem amb que la màquina està aixecada durant tot el curs, així que amb 240 hores (24 hores durant 10 dies, comptant el cap de setmana entremig) ens surt un preu total de 693€ per curs. Si tenim en compte els 10 cursos de mitjana a l'any, el cost total a final d'any serà de 6.930€, un 58% menys respecte el pressupost anterior.

Màquina activa durant les hores de curs
El cost de la màquina per hora és de 2,887€. Per aquest pressupost comptarem amb que la màquina només està aixecada les hores que s'imparteix el curs, és

a dir, 32 hores. El preu per curs seria de 92,4€, però comptant els 10 cursos de mitjana l'any surt un total de 924€ l'any, gairebé un 90% menys que el pressupost anterior.

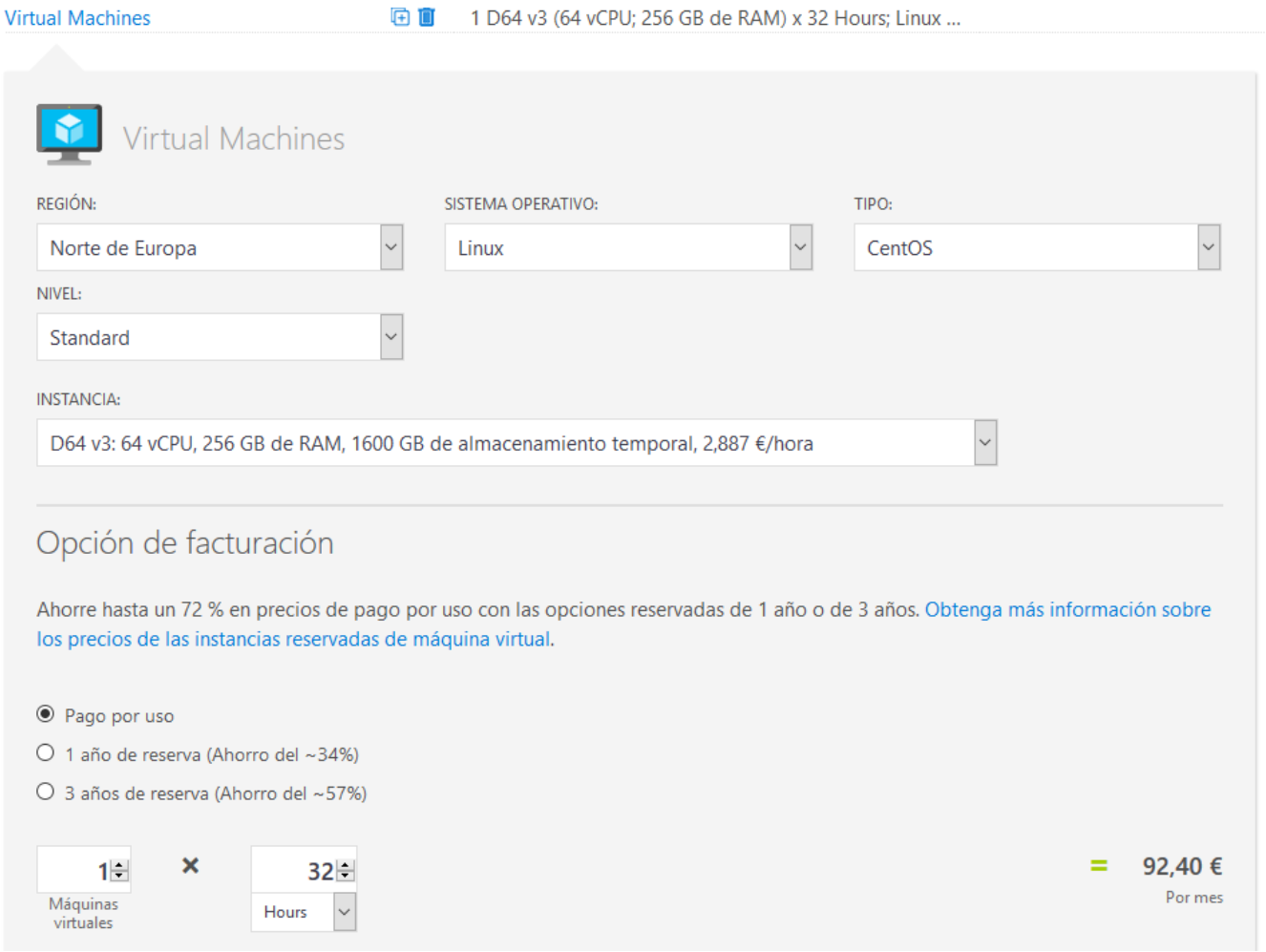


Figura 22: Calculadora de pressupostos d'Azure

Un cop analitzats els resultats dels pressupostos, es descartarà la primera opció de reserva d'un any. Aquesta reserva només permet estalviar si la màquina està activa durant 24 hores i 7 dies a la setmana, però no si només s'utilitza en determinades ocasions. Els dos pressupostos següents ajusten més el preu i serà l'InLab FIB

qui prengui la decisió final de quin pressupost s'adequa més a les seves necessitats. Per una part, la màquina activa durant tot el curs permet als alumnes accedir a la plataforma fora de l'horari del curs per practicar des de casa, però no és estrictament necessari perquè les pràctiques ja es fan en l'horari lectiu.

7 Gestió econòmica del projecte

7.1 Identificació dels costos

En aquest apartat s'estimaran els costos dels elements que s'han utilitzat per a aquest projecte. Aquests costos es classificaran de la següent manera: costos de recursos humans, costos de recursos tècnics (Hardware i Software) i costos generals. A part, es tindran en compte també els costos de contingència i d'imprevistos.

7.1.1 Costos en recursos humans

Per a aquesta estimació s'han comptabilitzat les hores de l'analista i del desenvolupador en una mateixa persona, ja que és el desenvolupador el que exerceix els dos rols. Les hores del Director de projecte s'han calculat tenint en compte el temps d'especificació i de seguiment del projecte. Finalment, respecte al personal de sistemes, s'han estimat les hores que han dedicat al suport al desenvolupador i al desplegament dels servidors.
El cost del personal inclou el salari brut i la Seguretat Social.

Recurs	Hores	Cost(€/h)	Cost total
Cap de projecte	40	60	2400€
Personal de sistemes	24	50	1200€
Desenvolupador	784	10.14	7949.76€
Total			**11549.76€**

Taula 3: Estimació dels costos humans

7.1.2 Costos en recursos hardware

Per als recursos hardware s'han estimat les hores i el cost tenint en compte el tems d'ús actiu dels dispositius. En el cas dels servidors, només s'han comptabilitzat les hores que han estat actius durant el projecte, encara que puguin seguir actius un cop

finalitzat.

Hardware	Cost unitari	Hores estimades	Amortització(€/h)	Cost estimat
Ordinador HP Compaq 8100	1200€	652	0.06	39.12€
Servidor DELL PowerEdge 2950	5569€	4320	0.31	1339.2€
Servidor DELL PowerEdge2950 Lirio	5591€	4320	0.31	1339.2€
Total				**2717.52€**

Taula 4: Estimació dels costos hardware

Per als recursos hardware s'ha tingut en compte 3 anys d'amortització i 6000h l'any laborables. També s'ha estimat un ús de 24h diàries per als servidors.

7.1.3 Costos en recursos software

Per la realització d'aquest projecte només s'ha utilitzat un software de pagament, tot i això es llistaran també els softwares gratuïts per tenir-los en compte en aquest projecte.

Per als recursos software s'ha tingut en compte 3 anys d'amortització i 6000h l'any laborables.

Software	Cost unitari	Hores estimades	Amortització(€/h)	Cost estimat
Windows 10 PRO	259€	600	0.043	25.8€
OpenNebula	0	0	0	0
VMWare	0	0	0	0
Latex OverLeaf	0	0	0	0
Toggl	0	0	0	0
Google Drive	0	0	0	0
Trello	0	0	0	0
Redmine	0	0	0	0
Total				**25.8€**

Taula 5: Estimació dels costos software

7.1.4 Costos generals

En aquesta estimació només s'ha tingut en compte allò que necessitava ser utilitzat exclusivament pel projecte. En relació a l'electricitat només es comptabilitzaran les hores i el consum de l'ordinador on es desenvolupava el projecte i les hores i el consum del servidor utilitzats per allotjar el sistema. S'ha comptabilitzat també, els costos i el temps del transport fins a la facultat, que és on es desenvolupa el projecte.

Per a aquesta estimació s'ha tingut en compte un consum d'uns 200Wh per l'ordinador i un consum de 510Wh per als servidors.

Recurs	Temps	Cost/temps	Cost total
Consum elèctric ordinador	652h	0,15 €/kWh	19.5€
Consum elèctric servidors	4320h	0,15 €/kWh	738.6€
Transport	8 mesos	80€/mes	640€
Total			**1398.1€**

Taula 6: Estimació dels costos generals

7.1.5 Contingència

Ens reservem una part del pressupost per a la partida de contingència. El nivell fixat per aquest projecte és de 15% sobre el total dels costos, tant directes com indirectes.

Recurs	Percentatge	Preu inicial	Cost
Costos directes	15%	15163.32€	2274.5€
Costos indirectes	15%	1398.1€	209.7€
Total			**2484.2€**

Taula 7: Estimació contingència

7.2 Imprevistos

Per calcular el cost dels imprevistos es tindran en compte alguns entrebancs numerats a l'apartat 5.5.

– Avaria dels servidors:

  En aquest cas, s'haurà de traslladar tot el sistema a un altre servidor. Això comportarà un cost en temps del desenvolupador i del personal de sistemes

(sense plantejar-nos la compra d'un servidor nou). Estimem una probabilitat del 5% que passi.

– Endarreriment en la implementació:

  Es podria donar el cas que el desenvolupador trigués més hores de les necessàries en finalitzar el projecte. Assumim un total de 15 dies d'endarreriment i una probabilitat del 15% a què això succeeixi.

– Caiguda del sistema:

  Tindrem en compte un endarreriment per la caiguda del sistema dels servidors utilitzats. Això comportaria una pèrdua de temps significant, ja que no hi ha còpia de seguretat i tant el desenvolupador com el departament de sistemes hauria d'encarregar-se'n. Es calcula un total d'1 setmana en restaurar el sistema i un 20% de probabilitats que es pugui donar el cas.

Imprevist	Probabilitat	Temps estimat	Cost per hora (€)	Cost final
Avaria	5%	60 h	12 desenvolupador i 50 sistemes	64.5€
Endarreriment	15%	60 h	12	108€
Caiguda sistema	20%	30 h	12 desenvolupador i 50 sistemes	148€
Total				**320.5€**

Taula 8: Estimació imprevistos

Per calcular el cost s'ha utilitzat la següent lògica: (Cost desenvolupador*hores + Cost personal sistemes*hores)*Probabilitat incidència.

7.2.1 Costos finals

Després d'haver estimat tots els costos i els imprevistos, calculem els costos totals del projecte per valorar la seva viabilitat econòmica.

Concepte	Cost final
Recursos humans	11549.76€
Recursos hardware	2717.52€
Recursos software	25.8€
Costos generals	1398.1€
Imprevistos	320.5€
Total	**16011.68€**

Taula 9: Costos totals del projecte

7.3 Control de gestió

El principal problema que pot sorgir en aquest projecte és la desviació temporal. Es podria donar el cas que les tasques triguessin més temps de l'establert en executar-se i això afectaria el pressupost del projecte. Per solucionar-ho, cada dia es fa una valoració sobre el que s'ha fet i el que es farà amb el Director de projecte, ja que treballem a la mateixa sala.

En el cas que sorgís algun imprevist que pogués afectar el procediment normal del projecte, aquest es comunicaria al Director de projecte i, si calgués, a la Cap de departament, per triar la solució més òptima i així perdre el menor temps possible.

Encara que no hi hagi cap entrebanc es quedarà amb la Cap de departament mensualment per fer el seguiment del projecte i comentar els avenços.

8 Sostenibilitat i compromís social

Seguidament es presenten 3 reflexions orientades al desenvolupament i la posada en marxa del projecte. Aquestes reflexions serveixen per conèixer la viabilitat econòmica del projecte i conèixer els impactes mediambientals i socials.

8.1 Reflexió econòmica

Per considerar si un projecte és viable o no, s'ha de realitzar una avaluació dels costos tant de recursos humans com hardware/software, inclosos també els possibles imprevistos que es pugui donar durant el projecte. Aquesta avaluació s'ha fet a l'apartat anterior (7). En el cas d'aquest projecte, veiem que els costos són bastant elevats, però tot i això segueix sent viable gràcies als beneficis que aporten els cursos.

Respecte a altres solucions com HackTheBox, aquest projecte (respecte a la posada en producció) serà més viable econòmicament, ja que només estarà actiu quan es facin els cursos que dóna l'InLab. En el cas que no hi hagi curs, els servidors estaran apagats i, per tant, no consumiran energia.

8.2 Reflexió mediambiental

L'impacte mediambiental d'aquest projecte actualment es força elevat. En ser un projecte en desenvolupament i en proves, es necessiten màquines engegades cada dia per poder realitzar les proves corresponents i desplegar el sistema constantment. A part d'aquestes màquines també fa falta un ús intensiu de l'ordinador on es desenvolupa l'aplicació, de manera que hi ha un consum energètic constat, però gràcies a que l'InLab disposa d'una aplicació per encendre i apagar l'ordinador de manera remota es fa més fàcil el control de consum i de temps d'ús. En el moment en què el projecte es desplegui en un entorn de producció el consum disminuirà dràsticament. El sistema està pensat per funcionar en períodes de temps curts (podria ser un parell o tres de dies al mes) on es realitzen els cursos i en finalitzar, les màquines podran estar apagades fins que es torni a necessitar el sistema pel següent curs.

Per a la realització d'aquest projecte s'han utilitzat els mínims recursos necessaris, tant pel desenvolupament com per les proves, de manera que és molt difícil disminuir l'impacte que té el projecte en la fase de proves i desenvolupament. L'única manera seria disminuint el consum elèctric dels servidors de proves, que estan encesos cada dia, però això comportaria una pèrdua considerable de temps, ja que s'haurien

d'encendre els servidors diàriament i podria fins i tot portar problemes amb la configuració d'aquests.

Respecte a altres solucions com *HackTheBox*, aquest sistema no estaria encès constantment, sinó que s'utilitzaria només quan fos necessari i durant períodes de temps curts, mentre que altres solucions tenen el sistema aixecat tot el dia.

8.3 Reflexió social

Com que aquest projecte ha estat demanat per l'InLab, és evident que hi ha una necessitat real que es vol resoldre. Primerament, permetrà estalviar temps al professor durant els cursos, facilitant la preparació del curs. Aquest projecte també permetrà descobrir i investigar noves tecnologies per l'InLab, que es podran aplicar en altres àmbits com el suport en les aules o en els laboratoris d'investigació. Alhora, és un projecte molt escalable que donarà oportunitats de treball a altres informàtics que mantindran i faran créixer el sistema.

Personalment, gràcies a aquest projecte he après molt sobre certs àmbits de la informàtica que mai havia tocat, com ara la virtualització o l'administració de sistemes. Un altre aspecte important és l'oportunitat que em dóna aquest projecte d'acabar el grau d'Enginyeria Informàtica i d'haver desenvolupat una plataforma que serà molt utilitzada.

9 Autoavaluació competència

Considero que la part mediambiental de qualsevol projecte pren un relleu molt important. S'hauria de tenir en compte sempre què és el que es pot fer per afavorir la reducció del consum d'energia o dels recursos.

En el cas d'aquest projecte, s'ha intentat reduir el màxim possible el consum energètic. Tot i tenir un servidor encès 24 hores al dia, aquest és compartit amb altres projectes dins de l'InLab, de manera que s'han estalviat molts recursos en utilitzar només un servidor per diversos propòsits. Com s'ha mencionat anteriorment, l'InLab disposa d'una eina (Tesla) que permet posar en marxa o apagar l'ordinador de la feina de manera remota. D'aquesta manera, si es necessita treballar amb l'ordinador de la feina des d'algun altre lloc, no cal deixar l'ordinador encès, sinó que es pot encendre o apagar a petició de l'usuari.

Com també s'ha comentat anteriorment, durant el desenvolupament del projecte, els servidors han estat engegats tot el dia, però en quan el projecte passi a producció, el servidor només estarà en marxa durant els cursos i això implica una reducció molt gran del consum, ja que en comptes d'estar 24 h en marxa ho estarà aproximadament 8 dies al mes.

10 Conclusions

10.1 Assoliment dels objectius

1. Configurar i integrar una aplicació web ✓
   Aquest objectiu s'ha complert gràcies a l'aplicació web que proporciona OpenNebula per administrar les màquines virtuals i configurar la plataforma.

2. Configurar un servidor per allotjar les màquines virtuals ✓
   Aquest objectiu no només s'ha complert, sinó que s'han configurat més d'un servidor, incloses proves de concepte amb servidors proporcionats per serveis *cloud* externs com Microsoft Azure.

3. Configurar, definir i introduir les màquines virtuals a la plataforma ✓
   Aquest objectiu s'ha complert configurant i introduint totes les màquines necessàries per a tots els tipus de curs que dóna l'InLab en ciberseguretat.

10.2 Treball Futur

Gràcies a l'escalabilitat d'aquest projecte, el treball futur no té gairebé límits. Tant la infraestructura com el número d'imatges a pujar a OpenNebula són totalment independents a la plataforma, de manera que es pot fer créixer el projecte en qualsevol de les dues direccions tant com es vulgui.

En el primer cas, es poden configurar tants servidors com hi hagi disponibles perquè facin de Node OpenNebula, però de moment, el futur més immediat és configurar un Node prou potent a Microsoft Azure per poder executar les màquines virtuals de manera fluida en relació a la quantitat d'usuaris utilitzant-les.

En l'altre cas, es poden pujar tantes imatges com siguin necessàries, de manera que sempre que es vulgui augmentar l'oferta de cursos es poden crear imatges especialitzades en el temari del curs i pujar-les a la plataforma.

10.3 Valoració personal

Considero que el desenvolupament d'aquest projecte ha estat satisfactori. S'ha implementat una plataforma totalment funcional, que millora qualitativament els cursos i que permet estalviar moltíssim temps tant als alumnes com al professor. En ser una plataforma *cloud* es pot accedir de manera remota, així els alumnes poden practicar des de casa sense la necessitat d'utilitzar els seus recursos, el que dóna moltíssimes facilitats.

Ha estat satisfactori també per la part personal, ja que aquest projecte ha estat l'últim pas per acabar el grau i poder entregar un treball final ben realitzat. Gràcies a aquest projecte també he aplicat moltes lliçons apreses durant la carrera i n'he après moltes de noves. He aplicat i augmentat els coneixements obtinguts del grau en l'administració de xarxes, l'administració de sistemes Linux, en ciberseguretat i he après també molts aspectes importants de la virtualització i el funcionament d'*hypervisors* i *orquestradors*.

Finalment, dir que gràcies a aquest projecte podrà venir algú altre i seguir desenvolupant-lo per així aprendre tant com he après jo.

Índex de figures

Índex de taules

Glossari

boot Procés en el qual s'encén un computador. Durant el temps de *boot* s'inicialitza el sistema operatiu i els dispositius connectats al computador. 36

CERT Computer Emergency Response Team. 47

cloud Conjunt de recursos compartits a la xarxa consumits com a serveis. 1–3, 13, 15, 24, 49, 50

CSIRT Computer Security Incident Response Team. 47

CSIRT Computer Security Incident Response Team. 19

forensics Aplicació de tècniques d'investigació i anàlisis per recollir i guardar evidències d'un dispositiu informàtic o sistema vulnerat o que ha patit un mal funcionament https://searchsecurity.techtarget.com/definition/computer-forensics. 45

hypervisor Software o *firmware* que s'encarrega de crear i executar màquines virtuals i de gestionar els recursos hardware de la màquina per distribuir-los entre les diferents instàncies. 16, 17, 19, 20, 27, 35, 36, 40, 63

IaaS Un tipus de *cloud* que proveeix recursos virtualitzats de xarxa i computació sota demanda, com servidors o xarxes virtuals. 13, 39

imatge Fitxer amb els continguts i l'estructura d'un volum de disc o d'un disc dur sencer. 31, 35, 45, 62

instanciar Crear una ocurrència concreta d'una màquina virtual i executar-la. 10, 12, 13, 19, 32, 34–36, 50

malware Software maliciós destinat a accedir o danyar un dispositiu de manera inadvertida sense el consentiment de l'usuari https://www.avast.com/es-es/c-malware. 46

malware reversing Procediment per extreure el codi font d'un *malware* per estudiar com funciona i quines accions pren sobre el teu sistema https://searchsoftwarequality.techtarget engineering. 46

orquestrador Software que permet la centralització de l'administració d'un centre de dades o *cloud* virtualitzat independent a la infraestructura. 9, 15, 16, 19, 24, 25, 30, 63

switch Dispositiu de xarxa que connecta diversos dispositius i permet la comunicació entre ells mitjançant la commutació de missatges (paquets) i l'adreça física de cada dispositiu. 40

Referències

[1] OWASP Top Ten Cheat Sheet, consultat el 16/03/2018
https://www.owasp.org/index.php/OWASP_Top_Ten_Cheat_Sheet

[2] Badstore 1.2.3, consultat el 28/02/18
https://www.vulnhub.com/entry/badstore-123,41/

[3] Metasploitable, consultat el 28/02/18
https://sourceforge.net/projects/metasploitable/

[4] PentesterLab - Web for pentester, consultat el 28/02/18
https://pentesterlab.com/exercises/web_for_pentester

[5] Hack this site!, consultat el 28/02/18
https://www.hackthissite.org/

[6] Hack The Box :: Penetration Testing Labs, consultat el 28/02/18
https://www.hackthebox.eu/

[7] OpenNebula - Flexible enterprise cloud made simple, consultat el 20/03/18
https://opennebula.org/

[8] Openstack - Open source software for creating private and public clouds, consultat el 25/03/18
https://www.openstack.org/

[9] vSphere - vSphere Hypervisor by VMWare, consultat el 25/03/18
https://www.vmware.com/es/products/vsphere-hypervisor.html

[10] Docker - Build, Ship, and Run Any App, Anywhere, consultat el 25/03/18
https://www.docker.com/